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

EXPERIMENTÁLNÍ SOFTWAROVÝ HUDEBNÍ NÁSTROJ ŘÍZENÝ ZMĚNOU INTENZITY SVĚTLA
EXPERIMENTAL SOFTWARE MUSICAL INSTRUMENT CONTROLLED BY CHANGE OF LIGHT INTENSITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Anna Laborová**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Audio inženýrství**
Ústav telekomunikací

Studentka: Anna Laborová ***ID:*** 174461
Ročník: 3 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Experimentální softwarový hudební nástroj řízený změnou intenzity světla

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat experimentální elektroakustický softwarový hudební nástroj; experimentálnost spočívá v netradičním způsobu ovládání parametrů zvuku změnou intenzity dopadajícího světla.

DOPORUČENÁ LITERATURA:

[1] PUCKETTE, M. Theory and Techniques of Electronic Music, 2006. 337 s. online: http://msp.ucsd.edu/techniques.htm

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

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 15.8.2018

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

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

Abstrakt

Práce představuje návrh a realizaci experimentálního softwarového syntezátoru *Ljós* užívající snímaní intenzity světla jako ovládacího prvku. Nástroj je realizován v prostředí programu Max/MSP a platformy Arduino, tělo ovladače je zhotoveno za pomocí technologie 3D tisku. Těžiště experimentálnosti spočívá v interaktivitě ovládání. Uživatel mění parametry syntezátoru pomocí světelného zdroje, konkrétně jeho intenzity.

Klíčová slova

syntezátor, světlo, syntéza, FM syntéza, aditivní syntéza, granulární syntéza, senzor, fotodioda, intenzita světla, Max/MSP, Arduino, multimédia, 3D tisk

Abstract

The goal of this document is to design and build an experimental software synthesizer *Ljós* using a light intensity as a trigger for changing the parameters. The instrument is programmed in the environment of Max/MSP and the controller is based on the platform of Arduino and made by 3D printer.
An user is changing parameters of the synthesizer by using a light source, by its intensity.

Keywords

synthesizer, synthesis, FM synthesis, additive synthesis, granular synthesis, light, light intesity, sensor, photodiode, Max/MSP, Arduino, multimédia, 3D print

LABOROVÁ, A. *Experimentální softwarový hudební nástroj řízený změnou intenzity světla*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 56 s. Vedoucí bakalářské práce doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D..

Prohlašuji, že svou bakalářskou práci na téma „Experimentální softwarový hudební nástroj nebo zvukový zdroj“ jsem vypracovala 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 jejím konci.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následku 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.

V Brně dne 15. 8. 2018 .....

podpis autorky

Poděkování

Ráda bych na tomto místě poděkovala především doc. Ing. MgA. Mgr. Danu Dlouhému, Ph.D. za vedení mé práce, cenné rady a připomínky a velkou vstřícnost. Dále poděkování patří Bc. Kristýně Uhrové za profesionální realizaci 3D modelu, Ing. Pavlovi Pokornému za vytištění těla ovladače na 3D tiskárně a celé své rodině za potřebnou pomoc a podporu.

Obsah

Úvod

Interaktivita a multimédia jsou zásadními tendencemi 21. století. I v oblasti hudby se setkáváme se stále alternativnějšími přístupy, např. doplňování hudby grafickými projekcemi či využívání virtuální reality při poslechu hudby. I k procesu vytváření hudby můžeme přistoupit jiným než tradičním způsobem vycházejícím z poslechové zkušenosti. Již od 20. let 20. století nacházíme příklady experimentování s jinými způsoby kompozice nebo vytváření zvuků, příkladem může být syntezátor *theremin* nebo počítačový systém pro generování hudby *UPIC*.

Přiložená práce si klade za cíl představit návrh a realizaci softwarového nástroje, který užívá měření světelné intenzity jako ovládací prvek. Zvuk je tak obohacen o další parametr – světlo. Idea vychází z analogie mezi zvukem a světlem. Výstupem je zhotovení ovládání, pomocí něhož je uživatel schopný vytvářet zvuk na základě své vizuální představy a vystoupit tak z konvenčního způsobu myšlení, osvobodit se od kompozičních předsudků a zvyků.

Základním principem nástroje je proces přeměny intenzity osvětlení na zvukový signál. Intenzita světla je snímána za pomoci fotocitlivých diod a na nich odečtené hodnoty jsou digitálně zpracovány v rámci platformy Arduino. Softwarová část nástroje je realizována v prostředí Max/MSP.

Syntezátor má na sobě dva nezávislé generátory. První generuje zvukový signál a druhý obálku, která je na signál uplatňována. Pomocí senzorů jsou kromě parametrů generátorů ovládány i další parametry – nastavení harmonického/neharmonického spektra, délka zrna pro granulární syntézu, efekty delay, pitch shifter a reverb, možnost náhodného posílání do levého nebo pravého kanálu sterea a bandpass filter.

Úvod mapuje kulturně-historické souvislosti a zásadní interaktivní nástroje a systémy, kterými se popisovaný nástroj inspiruje. Následující kapitoly jsou věnovány návrhu nástroje a samotné realizaci, kde je popsáno základní schéma a princip nástroje, část hardwarová a softwarová. Závěr shrnuje praktické využití nástroje.

1 Teoretická část

1.1 Multimédia

Hudba v kontextu multimédií se souvisle vyvíjí již dlouhá staletí. Důležitý je však samozřejmě „módní vliv“ a tak se tento vývoj v určitých historických obdobích střídavě vynořuje a zase mizí. Vzorovým příkladem je už např. koncept antického dramatu (počátky spojení slova a hudby v antických tragédiích). Tento koncept se znovu objevuje na konci renesance a vzniká tak barokní opera a o další dvě století později si ho přivlastňuje Richard Wagner jako odůvodnění své ideje Gesamtkunstwerku (souborného uměleckého díla). Ve 20. století se sklony k multimédiu opět vynořují a s nástupem moderny dochází k hledání nových a nových výrazových prostředků. Příchod postmoderny pak bourá veškeré hranice a umožňuje kompletní tvůrčí svobodu. Někteří teoretici tvrdí, že multimediální umění může být bráno jako zatím poslední slovo ve vývoji uměleckých disciplín. Jiní však soudí, že konce uměleckého vývoje v jednotlivých disciplínách již bylo dosaženo a nyní nastupuje vývoj napříč disciplínami (ruku v ruce s vývojem technologickým). [6]

Za základ multimédií můžeme považovat tendence hledat společného jmenovatele objektů vnímatelných různými smysly. Např. přiřazení barev tónům znala už čínská nebo starořecká kultura. V antické filosofii pak přichází s definicí „multimédií“ Aristoteles, který ve svém spisu *Peri psyché* (O duši) dělí skupiny vnímaných předmětů na předměty, které jsou každému smyslu vlastní (např. zrak-barva, sluch-zvuk), a na předměty společné všem smyslům (např. pohyb, klid, číslo). Právě tuto skupinu předmětů společných všem smyslům lze považovat za jeden z nejstarších pokusů o nalezení společné kategorie pro různé senzorické podněty. [2]

Vazba mezi hudbou a vizuální rovinou sahá daleko do historie a je jedním z nejrozšířenějších typů multimédia. Důvodem může být přirozený sklon představivosti při působení hudby na člověka, vyvolávání pocitů a vzpomínek. Vizuální ztvárnění je pak jednoduché předání tohoto působení dál bez nutného vysvětlování. Dříve zmíněné barvy přisuzované tónům jsou prvními projevy tohoto propojení a nevěnuje se mu pouze Aristoteles, ale i řada dalších hudebních teoretiků a fyziků napříč staletími (Gioseffo Zarlino, Athanasius Kircher, Luis-Bertrand Castel, J. W. Goethe, Isaac Newton, ad.). Vztahy barev a tónů se zabývali také skladatelé Nikolaj Rimskij-Korsakov, Alexander Nikolajevič

Skrjabin, Ivan Vyšněgradskij či Olivier Messiaen. Pro tuto generaci přelomu století bylo soustředění na otázky světla jako výrazového prostředku příznačné, často i klíčové. Světlo ji zajímalo především jako nositel subjektivních pocitů, nálad a stavů umělce. [7]

1.2 Alexander Skrjabin a synestezie

Významným dílem Alexandera Nikolajeviče Skrjabina (1872–1915) je *Prométhée, le poème du feu*, op. 60 (*Prometheus*, 1911), symfonická báseň pro orchestr, smíšený sbor a světelný klavír (příp. barevný klavír, *tastiera per luce* – klaviatura nástroje neslouží pouze k rozeznívání strun, ale současně ke spínání světel). V části *Luce* se měl celý koncertní sál ponořit do různých barev. Autor přiřadil tóniny kvintového kruhu k barvám duhy a taktéž jim atribuoval různé psychologické či fyzikální hodnoty. U Skrjabina se v případě barev a tónů jedná výhradně o emocionální asociace a vizualizace toho, co slyší. Světelná projekce pak neměla být pouze doprovodným prvkem k hudbě, ale měla jí být rovnocenným partnerem. [10]

Zmíněný jev schopnosti barevné vizualizace hudby, resp. barevného slyšení, se označuje *synestezie*. Nejedná se však pouze o spojení hudby a barev, pojmem *synestezie*, z řeckého *synaisthesis*, můžeme obecně označit spojené vnímání. Tento fenomén je mezioborovou záležitostí, zasahuje do neurologie, umění, psychologie i náboženství, a velmi úzce souvisí s multimediálním uměním. Označuje průběh a výsledek splynutí vjemů, které náležejí jednotlivým smyslům. Dnes je synestezie chápána především právě jako spojení hudby a barvy. Odhadem se týká zhruba jednoho člověka z 2000 osob. Avšak to, že si většina z nás spojuje jisté představy s vnímaným zvukem, nelze automaticky nazvat synestezií. [7]

Dnešní doba je z hlediska technologického vývoje jako stvořena pro synestetickou jednotu. Interaktivita je na vysoké úrovni a ještě ani zdaleka nebyly vyčerpány její možnosti. Multimediální prostředky jsou naprosto běžné, je však důležité se v těchto možnostech neztratit a neopakovat se, naopak objevovat nová spojení a nové významy.

1.3 Experimentální elektronické nástroje a interaktivní systémy

Vývoj elektronických nástrojů ve 20. století šel pochopitelně velmi rychle dopředu společně s vývojem techniky jako takové. Již na konci 80. let bylo možné využívat pro

generování, záznam, zpracování a reprodukci zvuku v reálném čase počítače. V 90. letech byly počítače používány pro záznam a editaci zvuku zcela běžně a často nahrazovaly hardwarové elektronické nástroje či efektové procesory. Na počátku 21. století softwarové nástroje prakticky začaly vytlačovat z trhu zvukové moduly, jejich použití je totiž velmi jednoduché a jejich cena se pohybuje v mnohem nižší kategorii. Jako vznikající problém se však ukázala absence vizuální složky, popř. živého interpreta. Požadovanými vlastnostmi, které měly být elektronické nebo elektroakustické hudbě navráceny, byly interaktivita a participace.

Z určitého úhlu pohledu lze považovat všechny elektronické nástroje svým způsobem za experimentální. Zvláště pak ty, které nevyužívají k ovládání tradiční mechanickou klaviaturu. Jedním z nich je **theremin**, monofonní nástroj navržený ve 20. letech 20. století sovětským fyzikem Lvem Sergejevičem Těrmenem (1896–1993). Ten si při práci ve Fyzikálně technickém ústavu všiml, že kapacita lidského těla může ovlivňovat frekvenci elektronkového oscilátoru. Tento jev je základem jednohlasého nástroje bez klávesnice, který tvoří tóny na základě pohybu rukou kolem něj – jedna ruka řídí výšku generovaného tónu, druhá sílu. Lev Těrmen sám se snažil o různé využití svého nástroje, v roce 1922 dokonce sestrojil *Illumovox*, přístroj s rotujícím optickým kolem, který po připojení k thereminu promítal měnící se barvu v závislosti na výšce hraných tónů. Mimo jiné ho zajímalo, jakou roli by mohl theremin hrát v taneční hudbě. Během svého pobytu v USA ve 30. letech pracoval i na projektech, kdy umístil thereminy do prostoru a nechal je reagovat na pohyby tanečníků. Nástroje pak spouštěly nejen zvukové paterny, ale i světelné. Theremin je s oblibou využíván dodnes, a stále je vnímán jako netradiční a experimentální nástroj. [4]

Dalším příkladem inovativního hudebního nástroje je **syntezátor ANS**. Jedná se o tzv. fotoelektronický nástroj, který byl navržen ruským inženýrem Jevgenijem Murzinem (1914–1970) mezi léty 1937–1957 a realizován v roce 1958. Syntezátor pracuje na základě světelného senzoru, který snímá světlo procházející přes otáčející se kola nebo posouvající se skleněnou desku. Název ANS je odvozen od iniciál ruského skladatele Alexandra Nikolajeviče Skrjabina, jehož umění se vyznačovalo syntézou různých složek[1]. [11]

[1] Viz kapitola týkající se A. N. Skrjabina dříve.

V 50. letech kromě Murzina pracovala se světlem ještě britská skladatelka elektronické hudby Daphne Oram (1925–2003), která mimo jiné je i první ženou, jež navrhla a zkonstruovala hudební nástroj. Je autorkou tzv. techniky **Oramics,** kdy využívala světlo k tvorbě elektronického zvuku. Techniku představila v roce 1957 (dále ji rozšířila v roce 1962). Spočívala v kreslení tvarů na deset 35mm filmových pásů uchycených ve velkém kovovém rámu. Muzikant pak kreslil tvary na pás, aby vytvořil jakési stínítko, kterým upravil světlo dopadající na pás, kde bylo zachyceno fotoelektrickými buňkami. [12]

Alvan Lucier (*1931) je americký skladatel a experimentátor na poli elektronické hudby, který v roce 1965 představil dílo *Music for Solo Perfomer*. Za pomocí EEG[2] elektrod připojených ke své hlavě detekoval impulzy vln alfa, kterých lze dosáhnout ve velmi relaxovaném stavu. Vibrace těchto vln mnohonásobně zesílil a následně je použil k rozechvění bicích nástrojů rozmístěných v prostoru. V současnosti se s podobným přístupem můžeme setkat u polského skladatele EAH Marka Chołoniewskeho (*1953), který ve svých multimediálních kompozicích (či spíše performancích) využívá speciálního ovladače-senzoru, jejž nasadí na hlavu. Snímané vlny jsou pak posílány do prostředí Max/MSP, kde jsou dále zpracovávány na slyšitelný audio signál. Chołoniewski se taktéž již od 80. let věnuje propojení hudby a světla navrhováním vlastních interaktivních systémů - např. MCOS (Multi Controller Optical Systém): počítačový systém, který konvertuje jakýkoliv pohyb ve světelném prostoru v komplexní zvukové struktury (např. kompozice *Lighting* /1995/, *dark&lightZone* /1999/, *Passage* /2001/); dále pak System LTTS (Light Time Triggering System): optický počítačový systém, který překládá změny světla v sérii počítačových kódů, které spouští hudební sekvence (např. kompozice *What You See You Get* /1989/, *Piękna i Bestia* /1991/, *Sha Ba Del'Mana* /1994/). [13]

Senzorické technologie od 50. let nabírají na své popularitě a jsou dále využívány v oblasti hudby a tance. Zásadním je společné dílo skladatelů Johna Cage a Davida Tudora a choreografa Merce Cunninghama *Variations V* (premiéra 1965, New York), které navázalo na myšlenku Lva Těrmena se spojením thereminu a tance. Tanečníci interagovali s analogovým zvukovým systémem, který sestával z thereminových antén a fotoelektrických buněk rozmístěných v prostoru, kde jednotlivé „zvukové zóny" byly vizuálně odděleny světelnými kužely. V České republice na toto dílo navazuje např. Jakub

[2] *Elektroencefalografie* (EEG) je diagnostická metoda používána k záznamu elektrické aktivity mozku. Povrchovými elektrodami jsou snímány změny v polarizaci neuronů.

Rataj s hudebně-tanečním představením *Umění manipulace* (premiéra 2014, v rámci tanečního festivalu Nová generace). Hudba se v tomto případě skládá z instrumentální a elektroakustické složky, která je tvořena zčásti předem nahranými samply a zčásti ovládána tanečníkem v reálném čase pomocí pohybového senzoru Hot Hand). [1]

Jedním z klasických příkladů audiovizuálního umělce v současnosti je francouzský skladatel elektronické hudby Jean-Michel Jarre (*1948). Jeho live performance se vyznačuje využíváním tzv. laserové harfy. Jedná se o elektronický nástroj (popř. MIDI ovladač), který v momentě, kdy dojde k přerušení paprsku, vyšle impulz a poté zazní zvuk. U nás se uměním spojujícím světlo a zvuk zabývá např. Jiří Suchánek (*1979), který se věnuje převážně stavbě interaktivních audiovizuálních instalací v přírodě nebo ve veřejném prostoru, nebo Tomáš Dvořák (*1978), známý též pod pseudonymem Floex, který pracoval např. na interaktivní instalaci Archifon 1,2&3, kdy návštěvníci pomocí laseru ukazovali na objekty, přičemž se následně na objektu objevila projekce a rozezněla hudba. [14] [15]

1.3.1 UPIC a další software transformující vizuální data na hudbu

Iannis Xenakis (1922–2011) byl skladatel, architekt a matematik řeckého původu, který většinu života působil v Paříži. Vnímal zvláštní spojení mezi architekturou a hudbou, hudbu označil jako „architekturu v prostoru“. Právě pohled architekta mu umožnil chápat hudbu jinak než v souladu s tradicí, a mohl tak překonávat dosavadní hranice. Xenakisovi poskytly zcela nové nástroje matematika a informatika. Hudbu vnímal jako organizaci zvukových entit a vztahů mezi nimi na základě logických operací. [3]

Významným Xenakisovým projektem je kompoziční nástroj **UPIC** (Unité Polyagogique Informatique de CEMAMu). Tento nástroj umožňoval uživatelům prostřednictvím tabletu kreslit, editovat a ukládat průběhy zvuku a jeho modulace, a vytvářet tak elektronickou počítačovou hudbu pomocí gest. Základem byl digitalizovaný tablet s vektorovým displejem připojený k počítači Hewlett-Packard. Osa X reprezentovala čas a osa Y frekvenci, uživatel pak mohl „kreslit kompozice“ přímo na tablet a počítač zajišťoval syntézu zvuku. Výsledné zvukové záznamy mohly být natáhnuty z několika sekund na hodinu, dále bylo možné zvuk transponovat, přehrát pozpátku nebo invertovat. Systém především umožňoval vše v reálném čase.

První verze počítačového systému pro generování hudby UPIC byla vytvořena v roce 1977 (CEMAMu, Paříž) a Xenakisovým záměrem bylo rozšířit ji do všech francouzských konzervatoří. Autor sám ho použil ve svých skladbách *La Legende d'Eer* (1977) nebo *Mycènes Alpha* (1978). Xenakis se tímto nástrojem přiblížil k myšlence skladatele Edgarda Varèse (1883–1965), který si přál „stroj produkující zvuk", jehož partitura by byla prováděna přímo, hudba by byla interpretována doslova tak, jak ji autor napsal. S novou verzí nástroje pak pracovali i skladatelé Jean-Claude Risset, Takehito Shimazu nebo Curtis Roads. [4] [16]

V současnosti na UPIC navazuje řada dalšího softwaru, který převádí informace vizuálního charakteru (obraz, video, intenzita světla, aj.) na zvukový signál či MIDI data. Tyto programy lze použít převážně post-produkčně, některé však nabízí možnost zpracování informací v reálném čase a tím umožňují zajímavou cestu ovládání hudebních parametrů za pomoci změny videa apod. Z estetického hlediska je v tomto případě spíše hodnocen samotný proces převodu (koncept, symbolika apod.), protože samotný převod obrazu nezaručuje stejně zajímavý hudební výsledek.

Jako příklad lze uvést programy HighC a IanniX. HighC je software, který umožňuje tvorbu zvuku za pomoci kresby. Je tvořen softwarovým syntezátorem, sekvencérem a mixerem. Převážně je používán jako pedagogická pomůcka pro studenty kompozice všech věkových kategorií. IanniX je grafický open-source sekvencér vytvořený v roce 2002 za podpory francouzského ministerstva kultury. Odkazuje se přímo na Xenakisův nástroj UPIC a k uctění jeho odkazu pro elektroakustickou hudbu je i po skladateli pojmenován. Program pracuje s myšlenkou, že v současnosti každý uživatel preferuje různé softwarové prostředí pro vytváření zvuku, a tak IanniX neobsahuje vlastní audio engine. Disponuje však synchronizací skrze Open Sound Control, a je tak schopný posílat nakreslené křivky a události v reálném čase do dalšího prostředí (např. Pure Data, SuperCollider, Max/MSP apod.). Prostředí sestává ze třech základních prvků – triggerů, které spouští jednotlivé události, křivek a kursorů, které se pohybují vymezeným prostorem a časem.

Od 12. století existuje vztah mezi hudbou a grafikou v podobě zápisu pomocí tradiční notace (systém 5 osnov, klíče, noty, pauzy, ad.). Ve 20. století se objevuje nový fenomén grafických partitur, díky kterým skladatelé byli/jsou schopni zapsat své hudební představy. Skladatel elektronické hudby Karlheinz Stockhausen (1928–2007) používal grafickou notaci, která zobrazovala frekvenci a časový průběh, inspiroval se tak

v systémech spektogramů používaných při zvukových analýzách (první partitura vydaná tiskem byla Stockhausenova *Elektronische Studie II.*, Universal Edition 1954). Grafické partitury mohou být z interpretačního hlediska a přesnosti provedení nejasné a složité, z výtvarného hlediska však mohou být naopak velmi zajímavé.

Principy grafických partitur se v podstatě uplatňují u softwarů, které převádí výtvarné dílo na hudbu. Převážně se jedná o piano-roll editory (editační okno MIDI událostí), které však disponují daleko většími možnostmi (různou tloušťkou a barvou linie je možné ovládat více parametrů najednou než pouze výšku, ad.).

Výhoda všech těchto uvedených způsobů je ta, že je velmi vhodná pro všechny začínající hudebníky a skladatele, zároveň pro ty pokročilé nabízí velký prostor, nové možnosti a svobodu kompozice. Rozšiřuje rovinu tvůrčího myšlení a zesiluje výsledný účinek. Tento koncept je označován jako grafická audio syntéza. Programy, založené na uvedeném konceptu, transformují vizuální informace na audio data (např. HighC, Coagula, aj.) nebo MIDI data (např. IanniX), nebo umožňují řízení hudebních dat kreslením. [27]

1.4 Zvuková syntéza

Zvuková syntéza označuje proces získávání určitého zvuku. Syntezátor je pak přístroj, který tento proces umožňuje. Pro syntézu elektronického zvuku je zásadní prvek generátoru různých signálů. Převážná většina oscilátorů elektronických nástrojů produkuje periodické signály, které jsou dále zpracovávány. Nejobvyklejšími generovanými signály jsou:

a) *sinusový signál* – ve spektru nejsou kromě základní složky obsaženy žádné další vyšší harmonické, filtrace tak na signál nemá vliv;
b) *trojúhelníkový signál* – spektrum obsahuje pouze liché harmonické složky, jejich amplituda rychle klesá;
c) *pilový signál* – spektrum obsahuje liché i sudé harmonické složky, jejich amplitudy klesají úměrně jejich pořadovým číslům, jedná se o harmonicky bohatý signál;
d) *obdélníkový signál* – spektrum závisí na střídě signálu. Pokud je střída 1:1, obsahuje signál pouze liché harmonické složky, pokud se střída odlišuje od hodnoty 1:1, obsahuje signál i sudé harmonické složky. Modulací šířky pulsů vzniká signál s proměnným spektrem.

Vedle periodických signálů se však můžeme setkat i s generátorem šumu – *NG* (Noise Generator), který produkuje signál s definovanými vlastnostmi. Nejvyužívanějším je signál *bílého šumu*, jehož energie je rovnoměrně rozložená v celém frekvenčním rozsahu. Kromě generování základního signálu oscilátory generují i signály modulační. Ty slouží pro tvorbu vibrata či tremola, rozlaďování a modulaci dalších parametrů. Průběh těchto signálů je zpravidla periodický. Produkovaná frekvence se pohybuje mezi desetinami a desítkami Hz, tyto modulační generátory se proto označují zkratkou *LFO* (Low Frequency Oscillator). [25]

1.4.1 Aditivní (součtová) syntéza

Základním a nejstarším typem syntézy je aditivní syntéza. V nejjednodušším případě vzniká součtem několika sinusových signálů s různými frekvencemi. Požadovaná zvuková barva se pak nastavuje vhodným poměrem amplitud jednotlivých signálů. Svým principem tak vychází z Fourierovy řady, kde jakýkoliv periodický signál dá rozložit na dílčí sinusové vlny o různých frekvencích a amplitudách. [25]

1.4.2 Subtraktivní (rozdílová) syntéza

Subtraktivní syntéza umožňuje v podstatě jen výše uvedené průběhy – sinusový, trojúhelníkový, pilový a obdélníkový. Její princip stojí na omezení harmonických složek původního signálu filtrem, velmi jednoduše jde tedy o oscilátor generující signál (pokud možno bohatý na harmonické složky) a za ním přidaný filtr. V procesu nevznikají žádné nové harmonické složky, některé naopak zanikají a některé jsou pouze zvýrazněny. Výsledný signál je vždy jednodušší než výchozí. [25]

1.4.3 Modulační syntéza

O modulační syntéze mluvíme tehdy, pokud je signál ze základního oscilátoru ovlivňován modulačním signálem z druhého oscilátoru. Vznikají tak nové harmonické složky úměrné součtům a rozdílům přítomných frekvencí. Frekvence obou signálů může být v čase proměnná, proto výsledné zvukové spektrum nemusí být pouze statické, ale taktéž se může vyvíjet.

Rozeznáváme dva základní typy modulační syntézy – frekvenční modulace, amplitudová modulace. Zvláštním případem amplitudové modulace je modulace kruhová, kdy dochází k úplnému potlačení původního signálu. Metoda modulační syntézy vede vždy k složitějšímu výstupnímu signálu, než byl signál výchozí. [17]

1.4.4 Tvarová syntéza (waveshaping)

Metody tvarové syntézy jsou založeny na přímé tvorbě průběhu signálu jeho nelineárním tvarováním nebo přímým grafickým zadáním průběhu. Metoda nelineárního tvarování je založena na zkreslení sinusového signálu při průchodu obvodem s nelineární charakteristikou. Nelinearitou vznikají nové součtové a rozdílové harmonické složky. Vlastnosti výsledného signálu jsou tedy závislé na průběhu tvarovací funkce, amplitudě vstupního signálu a frekvenčních poměrech sčítaných signálů.

K dalším metodám tvarové syntézy patří přímé zadávání časového průběhu signálu. Podstatou je možnost měnit tvar signálu přímo za pomocí grafického prostředí nebo proměnných parametrů. Nevýhodou této metody je téměř nulová kontrola výsledného zvuku. [17] [25]

1.4.5 Granulární syntéza

Granulární syntéza se řadí mezi tzv. slučovací metody syntézy. Principem těchto metod je využívání skládání malých zvukových segmentů dlouhých pár milisekund do větších celků.

V případě granulární syntézy je pro generování zvuku využito velmi krátkých časových výseků, tzv. granulí či zrn (délka se pohybuje v rámci 1–100 ms). Tyto granule jsou pak seskládány vedle sebe do celých bloků, jsou různě přeházeny a přehrávány velmi rychle za sebou.

Dle řazení granulí za sebou rozlišujeme granulární syntézu synchronní, kdy jsou granule řazeny ve stejném časovém odstupu za sebou a ve stejných lineárních vztazích mezi sebou, a granulární syntézu asynchronní, kdy jsou granule organizovány víceméně náhodně. Speciálním případem je granulární syntéza quasi-synchronní, kdy jsou granule řazeny za sebou v nepravidelných intervalech, které jsou náhodně generovány. [17] [25]

1.5 Max/MSP

Max je název označující grafický programovací jazyk a rozhraní. Jeho využití je velmi široké, nejvíce se s ním lze setkat v oblasti zvuku, videa a grafiky. Umožňuje vytvářet program graficky. Je dostupný pro operační systémy Microsoft Windows a Mac OS X. V Max/MSP lze zpracovávat data z řady rozhraní a přídavných zařízení a poskytuje tak nekonečný prostor pro interaktivitu mezi počítačovými procesy a vnějším prostředím.

Rozhraní bylo vytvořeno v 80. letech 20. století Millerem Puckettem (*1959) v pařížském institutu IRCAM (Institut de recherche et coordination acoustique/musique). Původně se jednalo o editor s názvem *Patcher*, byl pouze pro Macintosh a umožnil skladatelům nový způsob interaktivní kompozice. Jako první *Patcher* využil skladatel Philippe Manoury (*1952), který v roce 1988 napsal skladbu *Pluton* pro klavír a počítač, kde synchronizoval počítač s klavírem a díky tomu ovládal audio procesor Sogitec 4X. Od roku 1997 spadá MAX/MSP pod vývojářskou firmu Cycling '74. Pojmenování Max je odvozeno od jména průkopníka počítačové hudby Maxe Matthewse (1926–2011), který se v 50. letech významně podílel na vývoji hudebního softwaru. Již v roce 1957 (v době, kdy pracoval v Bellových laboratořích) vytvořil program MUSIC, první program pro generování zvuku. [21]

Jednotlivé programy se nazývají patche. Ty vznikají uspořádáním a propojením nejrůznějších bloků – objektů, zpráv, subpatchů a dalších. Objektům se funkce přiřadí vepsáním příslušného textu. Standardní balíček Max/MSP obsahuje kolem 600 těchto objektů, dostupných je několik rozšíření. Každý objekt disponuje vstupem a výstupem, což je základní princip předání informace mezi jednotlivými objekty. Pořadí provedených příkazů odpovídá grafickému uspořádání bloků v patchi.

Výsledný program může fungovat i jako stand-alone aplikace. Další možností je využití Maxu jako MIDI plugin softwaru pro DAW Ableton Live (rozšíření Max for Live).

Důležitou vlastností rozhraní Max/MSP je schopnost sériové komunikace se senzory, motory a dalšími komponenty za použití desky Arduino.

Paralelně s Max/MSP je dostupné open-source rozhraní Pure Data (Pd), jehož autorem je taktéž Miller Puckette a které je zdarma. Jedná se v zásadě o nekomerční verzi Max/MSP. [20]

Existuje řada dalších programů na obdobném principu. Software OpenMusic (OM) je vizuální programovací jazyk založený na jazyce Common Lisp od vývojářů IRCAM Music

Representations Research Group. Jedná se o prostředí vhodné především pro hudební kompozici a na rozdíl od softwaru Max/MSP nebo Pd, výstup je vždy zobrazen v konvenční hudební notaci, kterou je ještě možné skrze editor doupravit. Pro live processing je dostupné např. modulární prostředí Integra Live. Od principu Max/MSP se odlišuje tím, že uživatel nemusí psát kód nebo patchovat objekty. Software disponuje knihovnou vestavěných audio processing modulů, které mohou být jakkoliv propojeny, a to včetně jednotlivých parametrů, a sestaveny do bloků. Integra Live je vystavena na open source softwaru, její audio processing host využívá Pd engine, který uživatelům umožňuje nakódovat si v Pd vlastní moduly. Dalším softwarem vhodným pro live performance a sound design je např. AudioMulch vytvořený v 90. letech na základě performancí muzikanta Rosse Benciny, které byly charakteristické využíváním granulární syntézy v reálném čase. [22] [23] [24]

1.6 Arduino

Arduino je open-source vývojová elektronická platforma, založená na jednoduché počítačové desce a vývojovém prostředí, které slouží programování desky. Pomocí Arduina je snadné vytvářet interaktivní objekty. Je tak velmi vhodné pro vytváření multimediálního díla. Arduino deska získává impulzy od různých senzorů (např. snímač intenzity světla), a na základě těchto dat ovládá výstupy (např. spustí hudbu). Aby deska zpracovávala data podle požadavku uživatele, je nutné vytvořit program. K tomu je využito programovacího jazyka Arduino (založeného na jazyce *Wiring*) a Arduino softwaru (IDE), který je založený na prostředí *Processing*, což je knihovna pro jazyk Java, k níž je přidán vlastní editor. [9] [18]

Arduino samo o sobě na trh nepřineslo nic nového. Čím je však výjimečné, je fakt, že tato platforma je založena na sociální interakci. Kolem Arduina vznikla silná komunita lidí, kteří se díky jednoduchému a srozumitelnému prostředí mohou dostat k programování mikroprocesorů bez nutnosti znalostí složité architektury a programování celého systému. Arduino je z několika důvodů velmi vhodná platforma pro umělce a designéry. Důležitým faktorem pro nezkušené programátory je široká podpora komunity, kterou tvoří amatéři-nadšenci, ale i lidé z praxe s technickým vzděláním a zkušenostmi. Je tak snadné dostat se k informacím, návodům nebo zdrojovým kódům. Dalšími zásadními důvody je open-source projekt (volně dostupná schémata

apod.), dostupnost různých variant modulů a tím pádem možná flexibilita, připojitelnost a programovatelnost přes USB. Rozhodujícím je často jeho relativně nízká cena a dostupnost.

První návrh Arduina pochází z roku 2003, kdy Hernando Barragán pracoval na své diplomové práci na fakultě Interaction Design Institute Ivrea (IDII) v Itálii. Cílem bylo vytvoření jednoduchého a levného vývojového setu pro studenty a ulehčení práce umělcům a designérům využívajícím ve svých projektech elektroniku. Základními požadavky byla jednoduchost vývojového prostředí a jazyka, open-source software a open-source hardware. Diplomovou práci vedli Massimo Banzi a Casey Reas. Výstupem bylo zařízení *Wiring*. V roce 2005 se od komercializovaného projektu *Wiring* odpojili Massimo Banzi a David Mellis a založili vlastní projekt s názvem Arduino. V letech 2008–2014 se kvůli interním sporům projekt rozdělil na dvě větve – Arduino SRL a Arduino LLC. [8]

Idea open-source hardwaru (OSH) je stejná jako u open-source softwaru (OSS). U OSS je sdílen kód softwarového řešení. U OSH jsou volně dostupné všechny podklady potřebné k výrobě vlastního hardwaru.

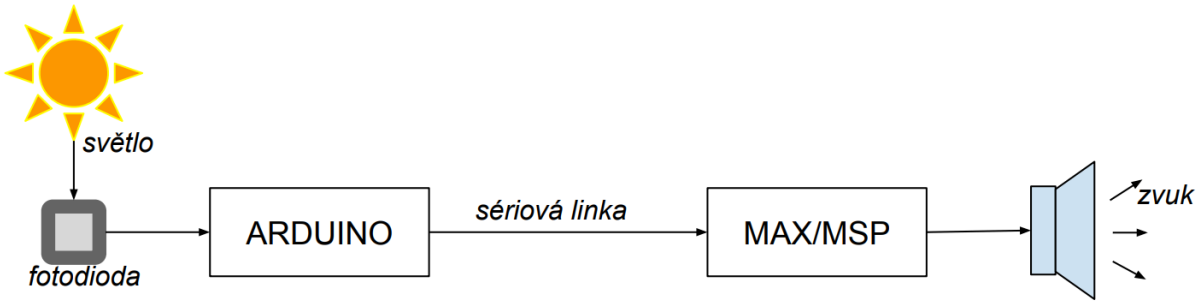
V současnosti existuje na trhu celá řada různých typů desky Arduino, od méně výkonných a malých modelů až po kompletní soustavy obsahující USB, HDMI, Ethernet či audio porty. Základem většiny desek Arduino je procesor od firmy Atmel. K nejmenším deskám patří např. Arduino Mini (nejmenší oficiální verze, absence USB portu, procesor ATmega328 s taktem 16 MHz, vhodný pro chytré vypínače, dálkové ovladače aj., 14 digitálních a 8 analogových pinů), Arduino Nano (téměř identický s Arduinem Mini, USB port, 22 digitálních a 8 analogových pinů) nebo *Arduino Micro* (čip obsahující převodník ATmega32u4, vhodné pro vytvoření vlastní klávesnice nebo herního ovladače, 20 digitálních a 12 analogových pinů). Standardním základním typem je deska *Arduino Uno*, které je v současnosti asi nejčastěji používaným typem. Obsahuje procesor ATmega328, USB port, 14 digitálních a 6 analogových pinů. Z tohoto modelu se vyvinuly další dvě speciální desky stejného rozměru – *Arduino Ethernet* (místo USB portu obsahuje ethernet port pro připojení k síti, slot pro microSD kartu) a *Arduino Bluetooth* (místo USB port obsahuje bluetooth modul pro bezdrátovou komunikaci). Zvláštním případem je model *Arduino Yún*, který sice designově navazuje na *Arduino Uno*, ale kromě čipu ATmega32u4 disponuje i čipem Atheros AR9331, který je schopen běhu odlehčeného operačního systému Linux OpenWrt-Yun. Komunikaci mezi oběma čipy zajišťuje

softwarový bridge (prostředník, most). Na desce lze nalézt USB port, microUSB port, Ethernet port, slot pro microSD kartu a modul WiFi. K největším deskám se řadí deska *Arduino Mega2560*. Jedná se v podstatě o prodloužení designu *Arduina Uno*. Je vhodná pro projekty, které vyžadují vyšší výpočetní výkon a větší počet pinů (54 digitálních a 16 analogových). Pokračovatelem tohoto modelu je *Arduino Due*, které běží na velmi výkonném čipu Atmel SAM3X8E (taktovací frekvence 84 MHz a 32bitové jádro oproti ostatním deskám s 8bity a maximálně 16 MHz). Na desce se nachází dva microUSB porty – jeden pro programování čipu, druhý pro připojení externího zařízení (např. myš, klávesnice nebo telefon). Díky tomu, že se jedná o open-source projekt, vznikala společně s oficiální řadou spousta dalších, neoficiálních desek, klonů. Často mají v názvu příponu „*–duino*“ (např. *Freaduino*, *Seeeduino*, *Rainbowduino*, *Richduino* a další). [9] [20]

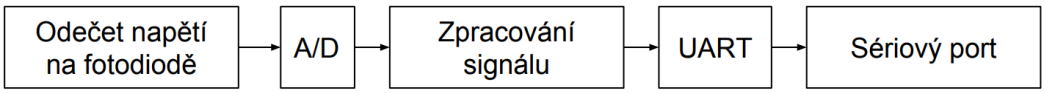
2 Návrh

2.1 Základní schéma

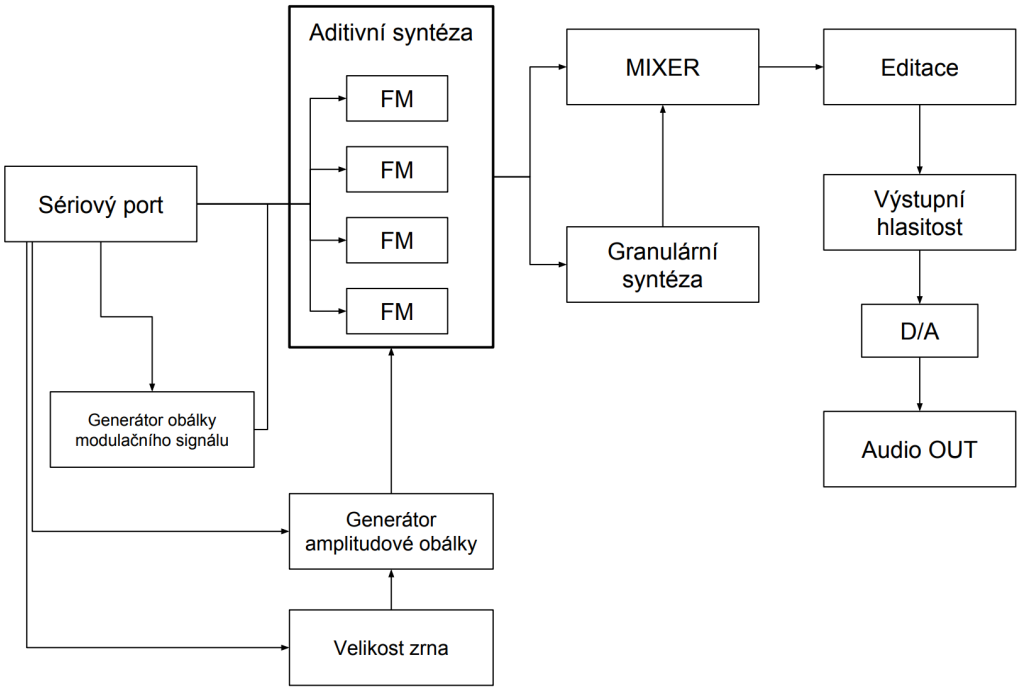
Podobně jako nástroj mechanický je i tento sestaven z několika základních funkčních bloků.



Obr. 2.1.1: Základní schéma syntezátoru



Obr. 2.1.2: Schéma bloku ARDUINO

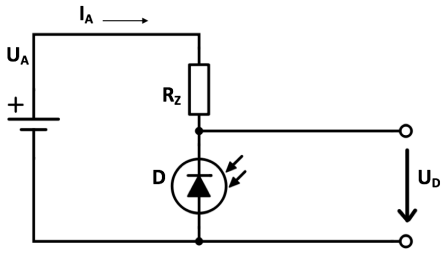


Obr. 2.1.3: Schéma bloku MAX/MSP

2.1.1 Hardwarová část

2.1.1.1 Fotodioda

Jako senzory jsou na vstupní analogové piny zapojeny fotocitlivé diody, na kterých je odečítána změna napětí. Fotodioda je polovodičová součástka konstrukčně upravená tak, aby do oblasti PN přechodu pronikalo světlo. Pokud tento přechod není osvětlen, má voltampérová charakteristika stejný průběh jako u běžné diody. Princip fotodiody je založen na vnitřním fotoelektrickém jevu. Dopadající fotony generují volné elektrony a díry. Vliv osvětlení můžeme sledovat v polarizaci diody v závěrném směru, kdy dochází k růstu proudu při zvětšování osvětlení. Dioda se tak chová jako pasivní součástka, jejíž elektrický odpor v závěrném směru je závislý na osvětlení. Fotodioda je zapojena v odporovém (fotovodivostním) režimu. Základními vlastnostmi tohoto zapojení je rychlá odezva na změnu osvětlení a velký poměrný rozsah výstupního signálu. Na změny osvětlení reaguje velmi rychle, řádově 10^{-6}–10^{-9} s. Proto je vzhledem k uvedeným vlastnostem tento princip používán k měření a detekci světla. [25]



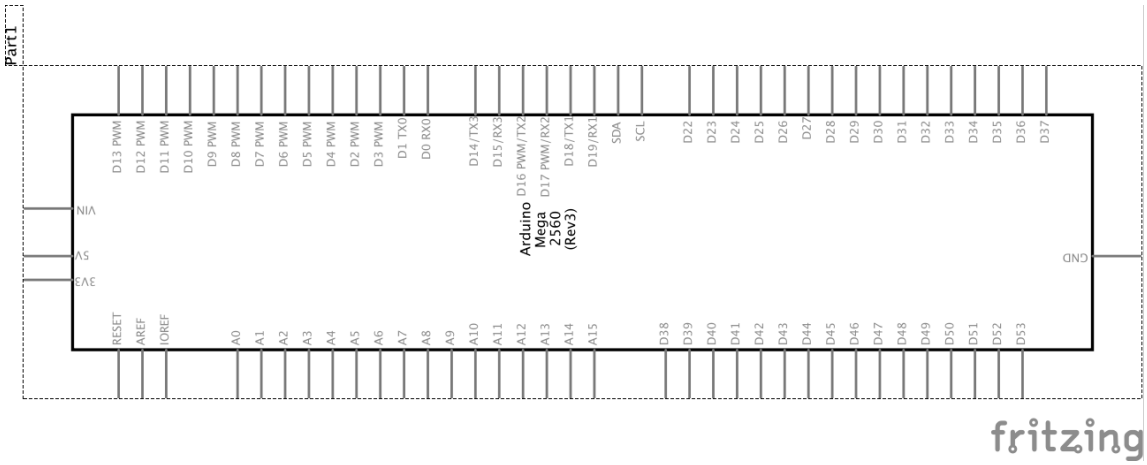
Obr. 2.1.1.1.1: Schéma zapojení fotodiody v odporovém (fotovodivostním) režimu; U_A – *napětí z desky Arduino,* I_A – *proud,* R_Z – *zatěžovací odpor,* D – *fotodioda,* U_D – *napětí na diodě*

2.1.1.2 Arduino

Z důvodu požadavku většího počtu zapojených fotodiod do analogových pinů, pro možnost ovládání co nejvíce parametrů, je pro projekt vhodná deska *Arduino Mega 2560* (16 vstupních analogových pinů) nebo *Arduino DUE* (12 vstupních analogových pinů).

Připojení desky k počítači je zajištěno jednoduše přes USB konektor typu B. Desku lze napájet přímo z počítače přes USB kabel, který zároveň zprostředkovává sériový přenos dat. [26]

Pro projekt jsou nejdůležitější analogové vstupy, na nichž jsou připojené vodiče měřící analogové hodnoty z jednotlivých senzorů.



Obr. 2.1.1.2.1: Schéma desky Arduio Mega 2560 [A]

Ke konverzi z analogových hodnot na digitální dochází v rámci procesoru ATMega2560 přímo na desce Arduino. Převodník má rozlišení 10 bit, pětivoltovou stupnici tak lze rozdělit na 2^{10} = 1024 hodnot (0–1023).

K naprogramování desky k požadovanému účelu měření a odesílání hodnot je potřeba *Arduino IDE*, vývojové rozhraní pro psaní programu. Po připojení desky k počítači je důležité toto rozhraní nastavit. V nabídce *Tools* a v seznamu *Boards* je nutno vybrat příslušnou desku (v tomto případě Arduino Mega 2560), poté v *Tools* -> *Serial Port* vybrat sériový port, na který je Arduino připojeno.

Při práci s analogovými hodnotami, které jsou odečítány ze senzorů, se používá funkce analogRead(pin), vracející hodnoty 0–1023, která ke své správné činnosti potřebuje uvést pouze jeden parametr analogRead(cislo_pinu). Tuto funkci je možné použít pouze na analogových pinech, které jsou označeny písmenem A.

Arduino defaultně měří na stupnici od 0 do 5 voltů (horní hranici lze upravit použitím pinu s označením AREF, referenčního napětí pro analogové vstupy, a funkce analogReference()). Naměřenou hodnotu je nutné uložit do proměnné pomocí promenna = analogRead(pin).

Princip komunikace desky Arduino s počítačem

Základním předpokladem je přítomnost USB portu na vybraném počítači. K propojení je třeba USB kabel. V případě převodníku je možné se setkat se třemi základními typy, které fungují stejně, liší se ale způsobem připojení.

Obr. 2.1.1.2.2: Schéma sériové komunikace

* Základní typy USB-serial převodníků:

1. Převodník napevno připájený k základní desce Arduina;
2. Převodník, který je obsažen přímo v některých čipech (např. ATmega32u4);
3. Externí převodník, který je nutné při programování k Arduinu připojit.

V případě Arduino Mega 2560 je převodník obsažen přímo v procesoru.

UART

UART je hardware na desce, který pomocí dvou pinů (označených jako RX a TX) odesílá a přijímá data. Jedná se o asynchronní způsob komunikace, přijímač i vysílač obsahuje vlastní generátor hodinového signálu, kterým se UART řídí. UART slouží ke komunikaci s dalšími externími zařízeními, jako je další deska Arduino apod. Piny RX a TX jsou však zároveň připojeny k čipu ATmega16U2 a tím pádem slouží jako sériová linka spojená s počítačem.

Knihovna Serial

Tato knihovna je standardní součástí vývojového prostředí Arduina, která používá hardwarový UART na digitálních pinech RX a TX.

Arduino komunikuje tedy pouze prostřednictvím sériového portu. Všechny funkce využívající sériovou komunikaci obsahují slovo Serial.

Ke čtení informací v textové podobě na počítači slouží tzv. *Serial monitor* (nabídka *Tools*, nebo ikona s lupou v pravém horním rohu IDE). Pro zobrazení číselných informací posílaných po sériové lince do grafu slouží funkce *Serial Plotter* (také nabídka *Tools*).

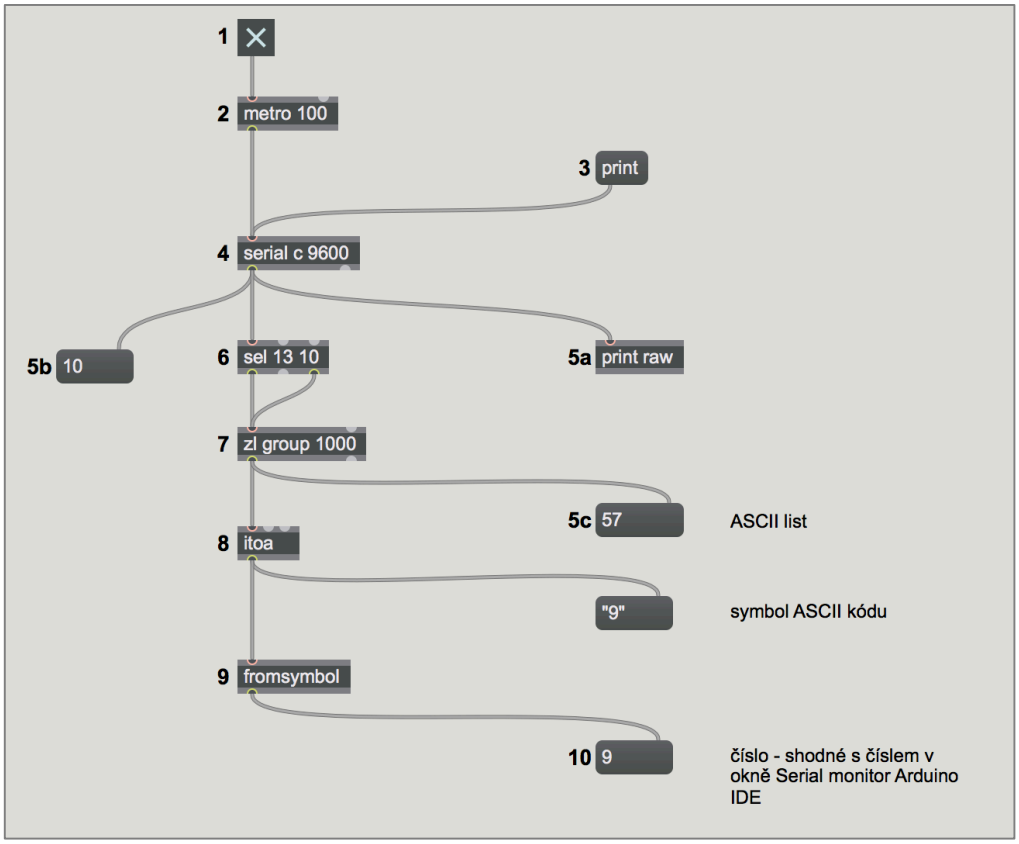
K zahájení komunikace se používá funkce Serial.begin(), která se volá v části setup. Do závorek je zadán parametr rychlosti komunikace, který odpovídá přesnému počtu přenosů za sekundu. Při práci s počítačem jsou k dispozici pouze tyto vybrané hodnoty: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 a 115200. Nejčastěji užívanou hodnotou je 9600.

K odeslání hodnot z Arduina do počítače slouží funkce Serial.print() a Serial.println(). Rozdílem je, že Serial.print() odesílá data stále na jednom řádku a Serial.println() přidává na konci odeslaných dat zalomení řádku. Jediným parametrem této funkce je odesílaná hodnota. Reálně se odesílá ASCII kód znaků a následně dojde v počítači k překladu, který je zobrazen v okně *Serial monitor*. [9]

2.1.2 Softwarová část

Základem výměny informací mezi deskou Arduino a rozhraním Max/MSP je opět sériová komunikace. V Arduino IDE je nejprve třeba napsat program pro čtení hodnot ze senzoru a posílat odečtené hodnoty pomocí funkce Serial.Print() na sériový port. Poté stačí program do desky nahrát, pro kontrolu bliká RX a TX LED dioda na desce a taktéž můžeme odesílané hodnoty sledovat v okně *Serial monitor*. Po nahrání kódu není nutné mít nadále aktivní okno programovacího rozhraní.

V Max/MSP bude základem práce objekt *serial*, který umožňuje posílání a přijímání dat do/ze sériového portu. Volitelnými parametry pro objekt *serial* jsou: port (symbol), název portu (symbol), rychlost komunikace (int), data (int), stop (int), parita (int). Pro tento projekt je zásadní název portu a rychlost komunikace.

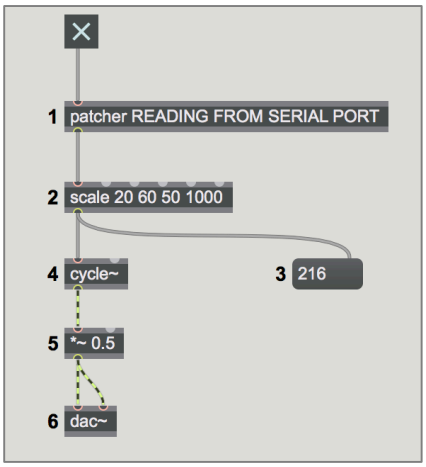


Obr. 2.1.2.1: Schéma kódu, který čte hodnoty ze sériového portu

1) *toggle* – spouští program, po každém kliknutí vysílá zprávu *bang* (neposílá žádnou hodnotu, ale vyvolává reakci dalších objektů), zároveň do výstupu odesílá hodnotu 1 při zapnutí a 0 při vypnutí;

2) *metro* – metronom, odesílá *bang* s nadefinovanou periodou 100 ms (popř. s periodou určenou hodnotou přivedenou do pravého inletu);
3) *print* – zpráva print vrátí seznam dostupných portů označených jako „port + písmeno (a,b,…)“, seznam se zobrazuje v okně *Max Console*;
4) *serial* – objekt, který přijímá data ze sériového portu, c = název vybraného portu, 9600 = rychlost komunikace, počet přenosů za sekundu;
5)
    a. *print raw* – MAX přijímá ze sériového portu tzv. raw data. Arduino načte informaci v podobě ASCII kódu. V Arduino IDE, v okně Serial monitor, jsou sice zobrazovány už přeložené hodnoty, MAX však ze sériového portu čte ASCII kód a je tedy nutné překlad naprogramovat. Objekt *print raw* vypíše odesílané hodnoty ASCII kódu do okna *Max Console*;
    b. zpráva vypisující pouze poslední přijímanou hodnotu ASCII kódu ze sériového portu, jako poslední dvě dvojčíslí se v seznamu objeví 13 a 10. Kombinace těchto dvou čísel reprezentuje konec řádku;
    c. zpráva vypisující kompletní ASCII kód;
6) *sel* (*select*) – čísla 13 a 10 představují oddělení jednotlivých ASCII kódů. Objekt *select* nám udává podmínku, která čísla budou dále zpracovávána. Argumentem 13 je dána podmínka, že pokud program obdrží číslo 13, pošle dál příkaz. Argumentem 10 je dána podmínka, že jsou vypsány všechny ostatní hodnoty, kromě 10;
7) *zl* – vytvoří aktuální skupinu všech hodnot jednoho ASCII kódu (argument 1000 představuje největší možnou velikost skupiny);
8) *itoa* – přeloží integer (popř. list až 256 integerů) v symbol (UTF-8 Unicode), nejedná se o výslednou použitelnou hodnotu;
9) *fromsymbol* – přeloží symbol v číslo. Jedná se o nejdůležitější část v kódu pro přijímání hodnot z Arduina;
10) zpráva vypisující číslo přeložené z ASCII kódu, shodnou hodnotu se zobrazenou hodnotou v okně *Serial monitor*.

Příklad generování sinusového signálu na základě hodnot ze sériového portu



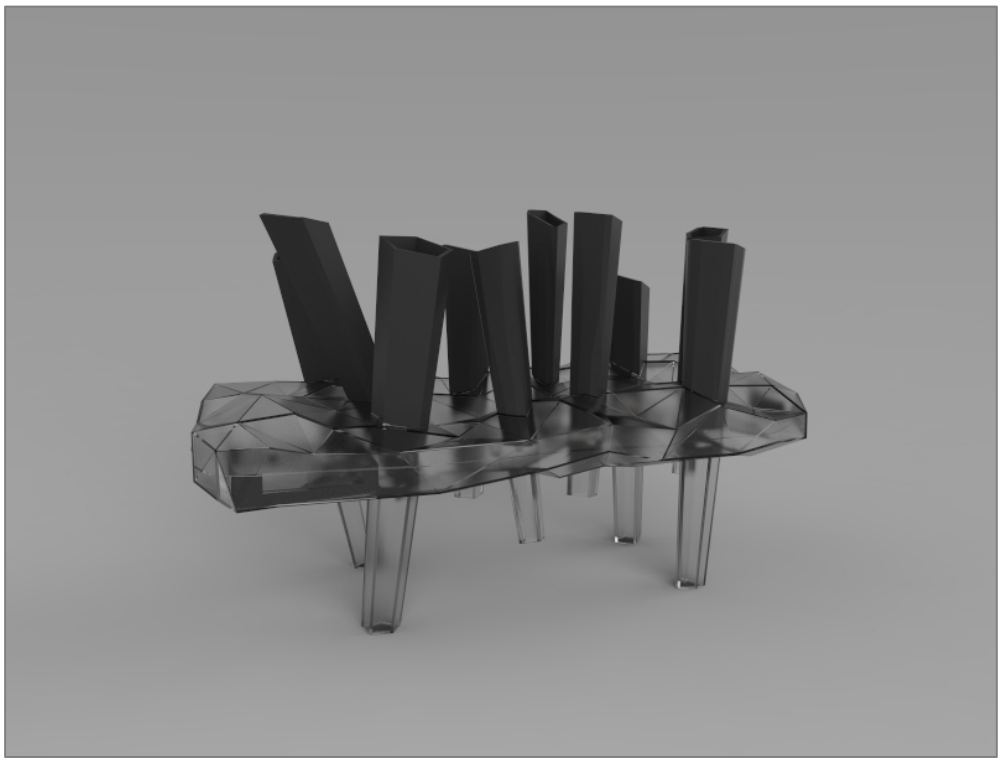
Obr. 2.1.2.2: Schéma kódu generujícího sinusový signál s frekvencí závislou na vstupních sériových datech

1) *patcher* – objekt vytvářející subpatch, v tomto případě obsahuje patch pro čtení dat ze sériového portu, viz obr. XY;
2) *scale* – umožňuje přepočet vstupních hodnot na vybranou stupnici nových hodnot. Parametry – nejnižší vstupní hodnota, nejvyšší vstupní hodnota, nejnižší výstupní hodnota, nejvyšší výstupní hodnota;
3) zpráva zobrazující výstupní hodnotu z objektu *scale,* odpovídá frekvenci v Hz;
4) *cycle~* – oscilátor generující sinusový signál, volitelným argumentem je frekvence, které je možné vepsat přímo za tildu, nebo přivést jako zprávu na levý inlet objektu. Přivedením zprávy na pravý inlet je resetována fáze;
5) *~ - násobení signálu
6) *dac~* - „digital to analog converter“ (D/A převodník), směruje všechny signály z Max/MSP do reproduktorů počítače, popř. externího audio zařízení.

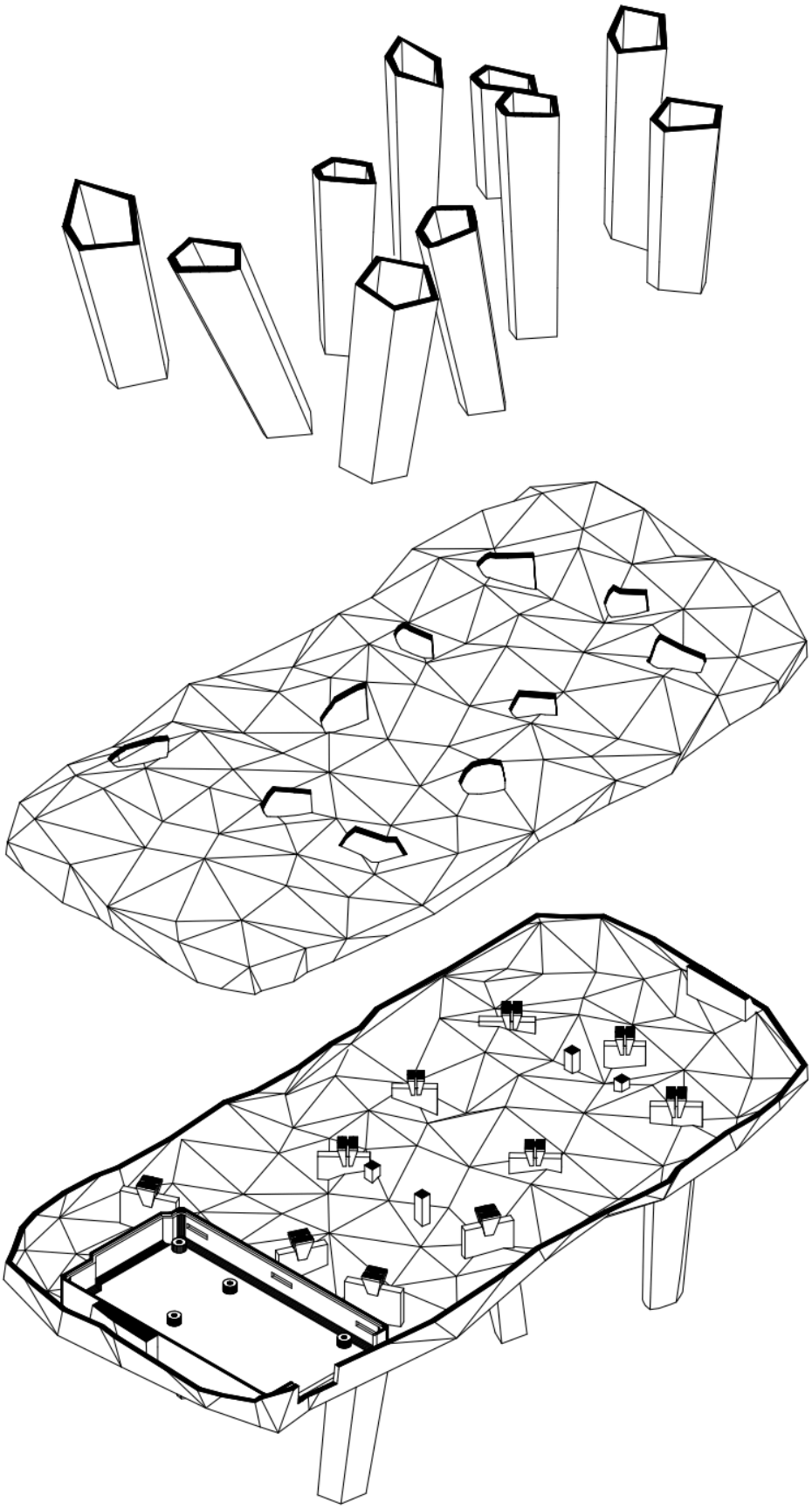
2.2 Design

Hardwarová část nástroje je umístěna ve speciálně navrženém těle. Digitální model je vytvořen v 3D modelovacím softwaru Rhinoceros a tělo je zhotoveno pomocí technologie 3D tisku. Design vychází ze struktur krystalu a ledovce, u kterých teprve světlo ukáže jejich skryté přednosti. Hráč svítí dovnitř černých krystalů, na jejichž konci je vsazena fotodioda. Černé stěny tak zabraňují propouštění světla k jiným senzorům. Spodní část těla je tvořena transparentním filamentem – v místech pod příslušnou fotodiodou je nástroj prosvěcován.

Ke hře je dále potřeba určitý světelný zdroj. Uživatel se může na základě vlastních potřeb a kreativity rozhodnout pro různé varianty – např. malé svítílny, které se do krystalů dají vsunout, či je možné sestrojit např. světelnou rukavici, která by měla na konečcích prstů LED diody (tato varianta ale může komplikovat manipulaci s přístrojem).



Obr. 2.2.1: Digitální 3D model ovladače



Obr. 2.2.2: Rozložený digitální model ovladače

3 Realizace

Finální podoba nástroje nese název *Ljós* (z islandštiny „světlo“), a to z důvodu, že tělo ovladače je inspirováno ledovcovými útvary ve Skandinávii. Hardwarová část prototypu (deska Arduino + pole s elektronickými součástkami) je vsazena do těla, které je vytištěno na 3D tiskárně. Softwarová část je zrealizována v IDE Arduino (program pro čtení hodnot z fotodiod) a v prostředí Max/MSP (mapování sériových dat na požadované hodnoty a zvuková syntéza).

Na vstupu softwarového rozhraní syntezátoru stojí sériová data, která jsou převedená z hodnot napětí odečtených na analogových (popř. digitálních) pinech desky Arduino.

Samotný syntezátor je založen na generátoru, ve kterém dochází ke složení aditivní a frekvenčně modulační syntézy. Čtyři oscilátory generují sinusový signál, který je následně modulován. Složení těchto čtyř signálů vytváří jednoduchou aditivní syntézu, na kterou lze uplatnit obálku, a vytvořit tak v reálném čase zvukovou granuli a tím granulární syntézu.

Základní frekvence prvního oscilátoru je závislá na datech prvního senzoru, další tři ladění vyšších harmonických složek jsou defaultně nastavena na celočíselné násobky základní frekvence, na druhý, třetí a čtvrtý. Může však dojít ke vzniku neharmonického spektra změnou parametru poměru nosné a modulační frekvence u FM syntézy. Tento parametr je ovládán druhým senzorem.

Třetí senzor ovládá bypass granulární syntézy. Uživatel je díky němu schopen plynule přejít z aditivní syntézy v granulární syntézu a s narůstající hodnotou světelné intenzity se zkracuje doba zrna, a výsledný zvuk se tak zahušťuje. Pokud tento senzor snímá hodnotu 0, nástroj nevydává žádný zvuk.

Oscilátory se rozkmitají velmi rychle a stejně tak rychle jejich oscilace po rozpojení obvodu zaniknou. Možnost nastavování obálky je tak více než žádoucí. Hodnoty ze čtvrtého senzoru ovlivňují obálku typu ADSR, specificky pouze parametr attack, a release. Uživatel je omezen na změnu těchto dvou parametrů současně, a to úpravou jejich poměru.

Obálka modulačních signálů je ovládána pátým senzorem. Změnou intenzity je určován čas parametru attack. Ostatní parametry zůstávají ve stejných poměrech, pouze se zkracují či prodlužují jejich jednotlivé délky.

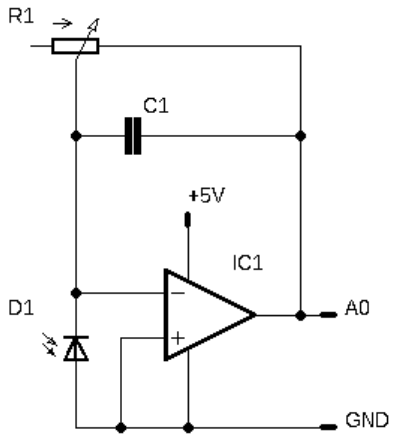
Součet signálů z oscilátorů a generátoru obálek je následně zesílen a poslán do efektové a filtrovací části.

Šestý senzor ovládá efekt *random pitch shifter*, který náhodně hýbe výškou tónu. Sedmý senzor udává čas pro zpožďovací jednotku delay, v rozmezí 30 až 1200 ms. Efekt je zde umístěn ke zjemnění zvuku granulární syntézy a jejího zahuštění. Osmý senzor pohybuje s mezní frekvencí filtru pásmové propusti. Devátý senzor aktivuje posílání granulí náhodně do levého či pravého výstupního kanálu. Dochází tak k prostorovějšímu zvuku. Desátý senzor nastavuje parametr trvání dozvuku.

V poslední části dochází k úpravě hlasitosti pro celkový výstupní signál. Signál je pak směrován do zvukové karty počítače (popř. externího zařízení), kde dojde k D/A konverzi a následnému zesílení.

3.1 Technické provedení nástroje

Hardwarová část nástroje je tvořena jednoduchým zapojením deseti fotodiod s deskou Arduino. Každá část s fotodiodou je doplněna o odpor, operační zesilovač a pro menší oscilace na výstupu je aplikován 10nF kondenzátor.



Obr. 3.1.1: Zapojení okruhu snímající intenzitu světl; D1 – *fotodioda,* IC1 – *operační zesilovač,* C1 – *kondenzátor,* R1 – *trimr,* A0 – *analogový pin 0,* GND – *země*

Místo odporu se stálou hodnotou je zvolen trimr s rozsahem 100 kW a to z důvodu, aby bylo možné nástroj přizpůsobit různým světelným podmínkám. Velikost odporu ovlivňuje citlivost fotodiody, a tak je tato možnost žádoucí.

Principem měření intenzity osvětlení je měření napětí na fotodiodě. Arduino měří s rozlišením 10 bit na stupnici 5 V, a tak je výsledná hodnota vracena v rozpětí 0–1023[3]. Pro účely nástroje není nutné přepočítávat získanou hodnotu na napětí. Je toho však popřípadě možné dosáhnout jednoduchým výpočtem:

```
int pinRead0 = analogRead(pinNo);
float pVolt0 = pinRead0 / 1024.0 * 5.0;      // převod na napětí
```

Kód pro měření intenzity čte vstupní hodnoty na analogových pinech, pro stabilnější výsledek je přidáno průměrování z pěti hodnot (vyšší počet prodlužoval dobu trvání programu a způsoboval pomalou odezvu senzoru).

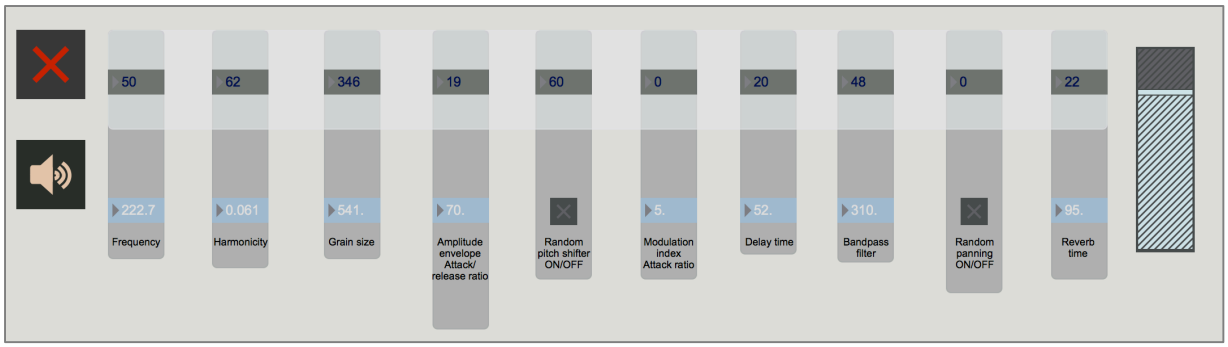
Kompletní kód pro čtení hodnot ze senzoru je dostupný na konci dokumentu jako příloha *A Program pro čtení hodnot z analogových vstupů Arduina*.

Obr. 3.1.2: Rozložený kontroler Ljós

[3] Viz kapitola 2.1.1.2 Arduino, str. 24.

3.2 Struktura projektu v Max/MSP

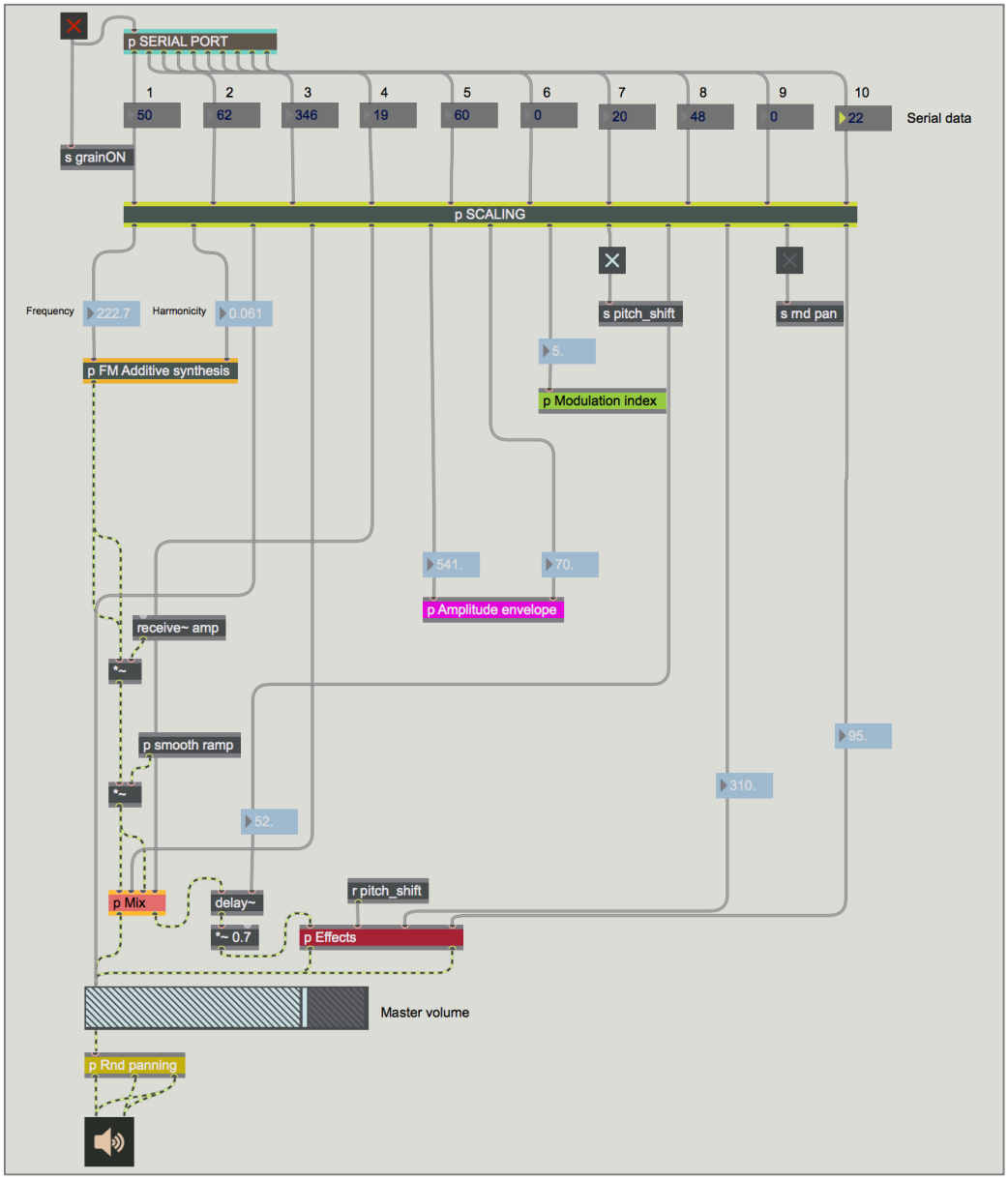
- Main patch
  - Serial port
  - Scaling
  - FM Additive synthesis
    - FM generator
  - Envelope generator
  - Modulation index
  - Mix
  - Effects
  - Rnd panning



Obr. 3.2.1: GUI syntezétoru Ljós – presentation mode Max/MSP

3.3 Jednotlivé patche projektu

3.3.1 Main patch

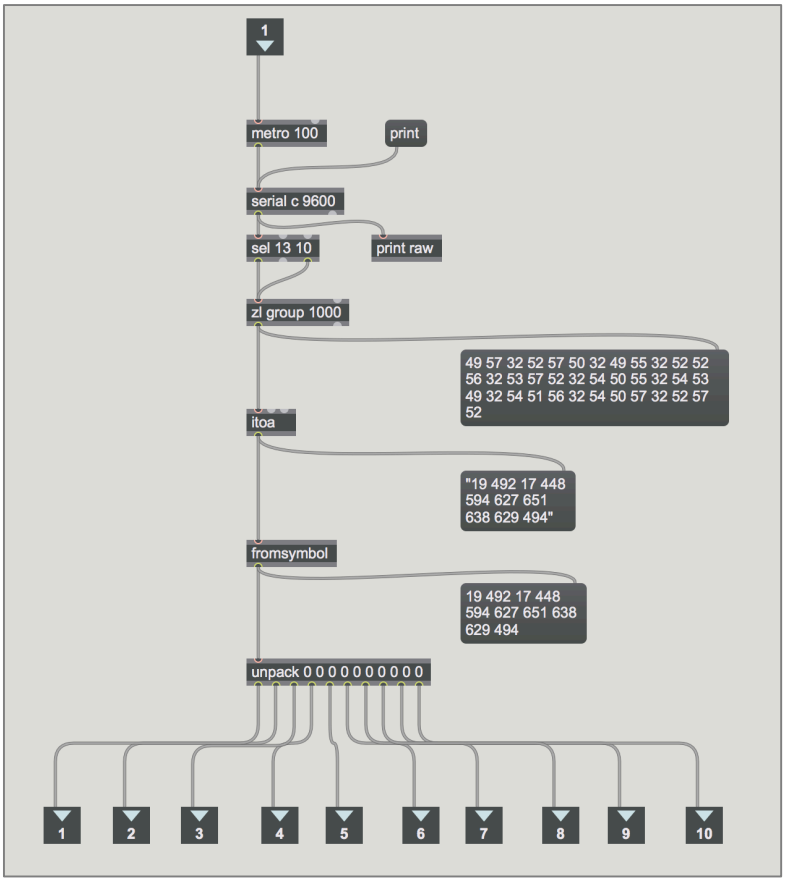


Obr. 3.2.2: Detail hlavního patche

Hlavním tlačítkem *toggle* dojde ke spuštění čtení hodnot ze sériového portu a ke generování amplitudové obálky a obálky modulačního signálu. Hodnoty prochází skrze subpatch *Serial port* do subpatche *Scaling*, kde je přehodnocen na požadované stupnice. Z něj jsou přeložené hodnoty dále posílány do generátoru v subpatchi *FM generator*, kde dochází ke složení aditivní syntézy ze čtyř signálů FM syntézy a dalších subpatcherů představujících směnu určitého parametru (podrobněji viz následující kapitoly o jednotlivých supatchích). Výsledný signál je poslán přes výstupní hlasitost do objektu

Rnd panning, kde může volitelně dojít k náhodnému rozložení jednotlivých zvukových granulí do sterea. Signál je pak směrován do objektu *ezdac~* a z něj do zvukové karty počítače či externího zařízení.

3.3.2 Subpatch Serial port

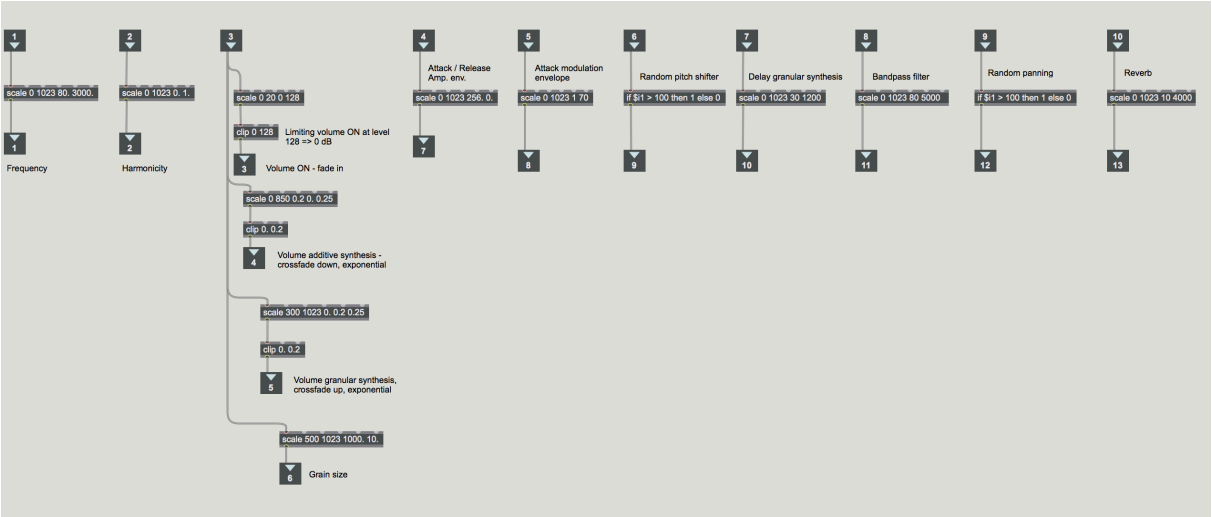


Obr. 3.2.3: Detail subpatche Serial port

Subpatch *Serial port* představuje základní patch projektu. V této části dochází k přijímání sériových dat do Max/MSP. Kód je podrobněji již popsaný v kapitole věnující se v návrhu,[4] zde je doplněn však o důležitý objekt *unpack*, který od sebe oddělí přečtené hodnoty z jednotlivých deseti senzorů, jelikož hodnoty jsou přijímany jako řady po 10 číslech oddělených zalomením řádku.

[4] viz kapitola 3.1.3 Max/MSP, str. 28

3.3.3 Subpatch Scaling



Obr. 3.2.4: Detail subpatche Scaling

V této části dochází k namapování snímaných hodnot ze senzoru na hodnoty vhodné na požadovaný parametr. Jak již bylo v textu dříve vysvětleno,[5] k tomuto účelu slouží objekt *scale* – umožňuje přepočet vstupních hodnot na vybranou stupnici nových hodnot. Na jednotlivé inlety jsou tedy postupně přiváděny hodnoty ze sériového portu a z outletů jsou posílány převedené hodnoty. První senzor ovládá základní frekvenci nástroje v rozsahu 80–3000 Hz.

Druhý senzor ovládá parametr poměru frekvenční a modulační frekvence u základní FM syntézy. Ke slyšitelné změně musí být zároveň aktivován senzor ovládající obálku modulačního signálu.

Třetí vstup představuje bypass granulární syntézy a tím možnost přechodu od konstantního tónu přes pulzující tón až k hustému zrnitému zvuku. Přechod je vyřešen mixem výstupů z aditivní a granulární syntézy, kdy mezi hodnotami 0–50 dochází k postupnému fade in celkové hlasitosti, od hodnoty 50–850 dochází exponenciálně k poklesu hlasitosti z výstupu aditivní syntézy, a mezi hodnotami 300–1023 exponenciálně narůstá hlasitost z výstupu granulární syntézy. Exponenciální průběh je určen pátým volitelným argumentem objektu *scale*. V momentu, kdy intenzita osvětlení dosáhne hodnoty 500, začne senzor současně ovládat i velikost zrna, a to postupným zkracováním.

Hodnoty ze čtvrtého senzoru ovlivňují procentuální část parametrů attack a release amplitudové obálky typu ADSR. Uživatel je omezen na změnu těchto dvou

[5] viz kapitola 3.1.3 Max/MSP, str. 30

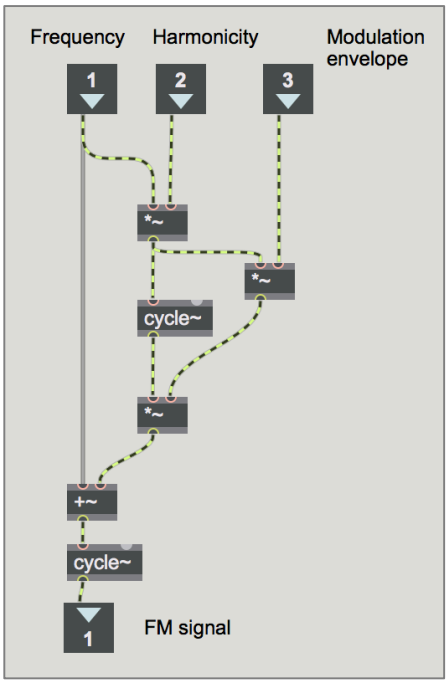
parametru naráz, zajímavým řešením je však změna těchto parametrů – když se doba attacku zvětšuje, doba release se zmenšuje a naopak. Doba je omezena na maximálně 70 % celkové doby trvání zrna.

Změnou intenzity na pátém senzoru je určován poměr parametru attack. Doba je nastavena na 1–70 % z celkové doby trvání zrna. Ostatní parametry zůstávají ve stejných poměrech, pouze se zkracují či prodlužují jejich jednotlivé délky v závislosti na změně attacku. Více viz kapitola Subpatch Modulation index.

Šestý senzor je v režimu tlačítka, hodnoty menší než 100 vrací 0, všechny ostatní vrací 1. Ovlivňuje spuštění efektu *random pitch shifter*, který náhodně hýbe frekvencí výstupního signálu granulární syntézy. Obdobně pracuje devátý senzor s tím rozdílem, že spouští efekt *random panning*.

Sedmý senzor mění čas delaye, v rozpětí 30–1200 ms. Osmý senzor ovlivňuje mezní frekvenci pásmové propusti od 80 Hz do 5 kHz. Desátý senzor ovládá čas dozvuku v hodnotách 10–4000 ms.

3.3.4 Subpatch FM generator

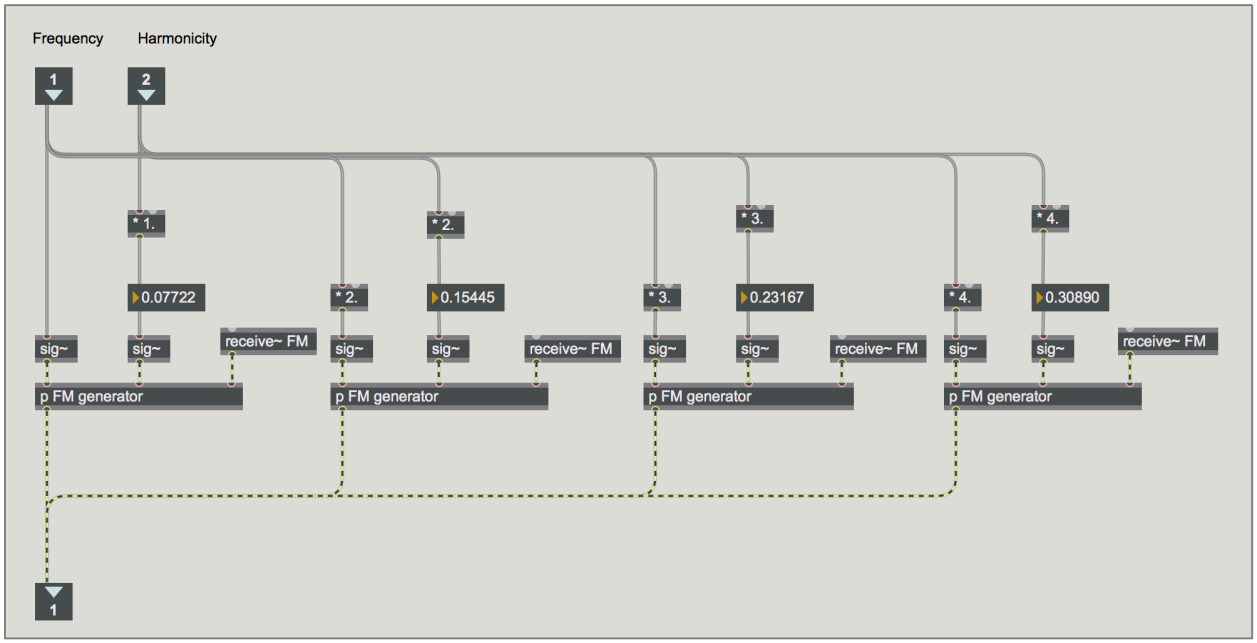


Obr. 3.2.5: Detail subpatche FM generator

V této části dochází ke generování základního signálu. Jedná se o jednoduchou frekvenčně modulační syntézu. Na prvním inlet je přivedena základní frekvence, hodnotami z druhého je ovlivňován poměr mezi základním a modulačním signálem a třetím inletem vstupují hodnoty ovlivňující hloubku modulace. Frekvence a poměr signálů jsou

konstantami, hloubka modulace je určena v čase proměnným signálem z generátoru obálky modulačního signálu.

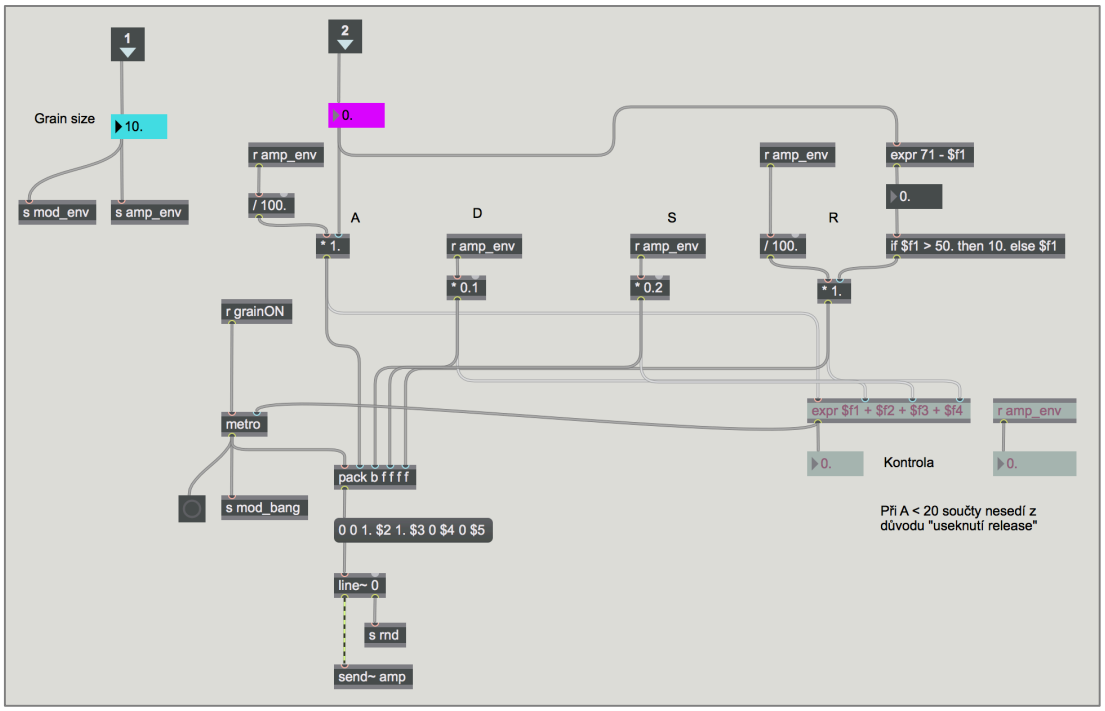
3.3.5 Subpatch FM Additive synthesis



Obr. 3.2.6: Detail subpatche FM Additive synthesis

Subpatch *FM Additive synthesis* obsahuje čtyři generátory FM syntézy a dochází k jejich sčítání. Do prvního generátoru je posílána základní frekvence ovládána prvním senzorem, další jsou celočíselnými násobky, druhým, třetím a čtvrtým. Na druhý inlet jsou přivedeny hodnoty z druhého senzoru a mění poměr modulačního signálu v různých poměrech. Objektem *receive~ FM* je přijímána obálka ze subpatche *Modulation index*.

3.3.6 Subpatch Amplitude envelope



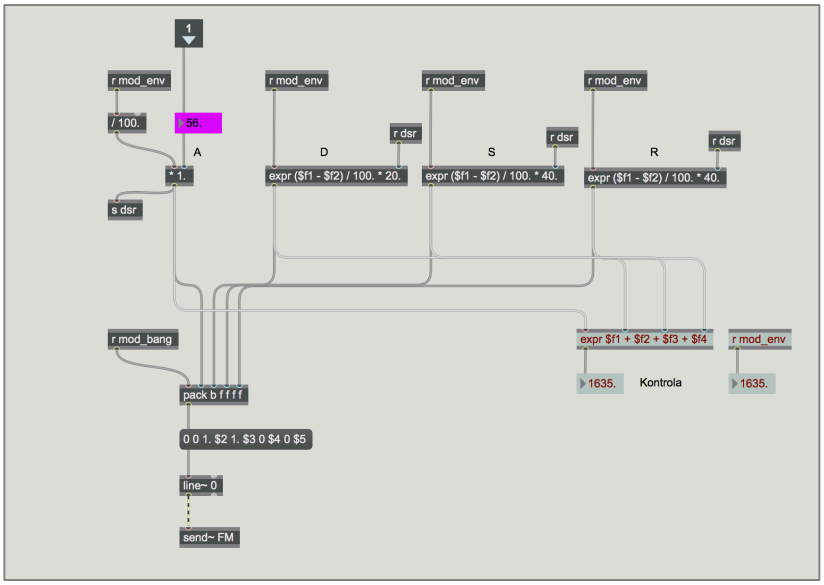
Obr. 3.2.7: Detail subpatche Amplitude envelope

Subpatch *Amplitude envelope* představuje generátor amplitudové obálky, která je aplikována na generovaný zvukový signál a v reálném čase vytváří zrna pro granulární syntézu. Vstupem jsou hodnoty parametrů attack, delay, sustain a release, přičemž intenzitou světla jsou ovládány pouze parametry attack a release, a to jejich poměr. V momentu, kdy se zvyšuje doba attack, dochází ke snižování doby release. Přidána je zde podmínka, že pokud je doba attacku menší než 20 % celkové délky obálky, je release náhle zkrácen na 10 %, aby bylo dosaženo velmi krátkých pulsů.

Při experimentování se parametry attack a release ukázaly, že mají na výslednou barvy zvuku granulární syntézy největší vliv, proto jsou ovládány právě tyto, a s ohledem na ovládání, je to zároveň jednodušší cesta, jak dosáhnout efektní změny. Uživatel je schopen docílit velmi hustého zvuku tvořeného velmi krátkými zrny, které zní jako pouhé clicky. Parametry decay a sustain zůstávají ve stejných poměrech, decay 10 % a sustain 20 % celkové doby trvání jedné obálky. Doba mezi jednotlivými zrny je závislá na délce trvání jednoho zrna – to je zajištěno objektem metro, na jehož pravý inlet je přivedena hodnota odpovídající aktuální délce zrna, která je v reálném čase měněna hodnotami ze třetího senzoru. Ten ovládá zároveň i délku trvání obálky modulačního signálu. Objekt *pack* „zabalí“ hodnoty jednotlivých parametrů obálky a pošle je přes *message* do objektu *line~*, který zajišťuje lineární průběh obálky. Z levého outletu je obálka poslána k násobení

se signálem ze subpatche *FM Additive synthesis*. Pravý outlet vysílá *bang* po skončení jedné obálky, v projektu je této vlastnosti využito pro efekt random pitch shifter a random panning.

3.3.7 Subpatch Modulation index



Obr. 3.2.8: Detail subpatche Modulation index

Na inlet č. 1 jsou přiváděna data z pátého senzoru, přeložena na hodnoty 1–70. Změnou těchto hodnot dochází ke změně času parametru attack. Může se měnit v 1–70 % času celkového trvání doby zrna a ostatní parametry decay, sustain a release jsou na něm závislé, jejich poměry se však dále nemění.

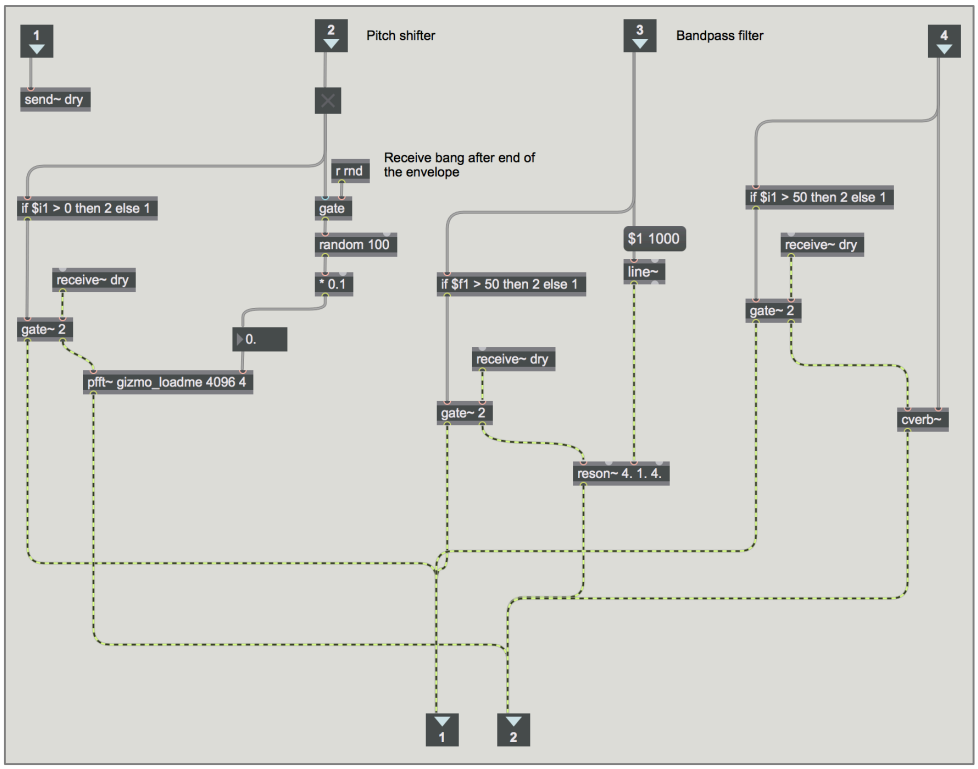
K výpočtu jsou aplikovány vztahy:

$$\begin{aligned} t_{DSR} &= t_{ADSR} - t_A, \\ t_D &= \frac{t_{DSR}}{100} \cdot 20, \\ t_{S,R} &= \frac{t_{DSR}}{100} \cdot 40, \end{aligned} \tag{1.1}$$

kde t_{ADSR} představuje dobu celkového trvání jedné obálky, t_{DSR} dobu součtu parametrů decay, sustain a release, t_{A} dobu trvání parametru attack, t_{D} decay, t_{S} sustain a t_R release.

Objekt *expr* umožňuje jakékoliv matematické operace s čísly přivedenými na jeho inlety, je však důležité zachovat v rovnici zápis jejich pořadí (první inlet $f1, druhý inlet $f2). Přes objekt *pack* jsou hodnoty posílány do *message*, která aktivuje objekt *line~* zprostředkovávající lineární obálku.

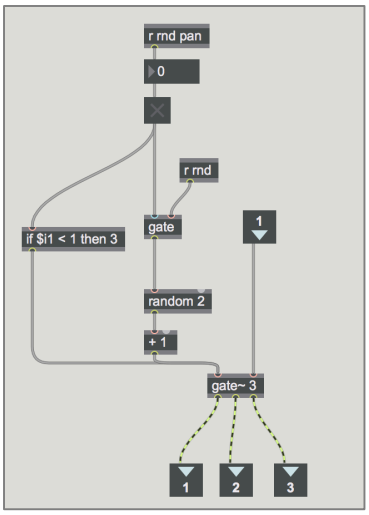
3.3.8 Subpatch Effects



Obr. 3.2.9: Detail subpatche Effects

Na první inlet je přiváděn neprocesovaný signál, který může být zpracován efektem pitch shifter, reverbem nebo filtrován pásmovou propustí. V části *Pitch shifter* dochází k aktivování efektu v objektu *pfft~*, který náhodně hýbe s frekvencí vstupního signálu. Efekt se sepne po dosáhnutí hodnoty 100, do té doby je od něj signál odkloněn. Objekt *random* generuje sto náhodných hodnot v rozsahu od 0 do 10. Při experimentování vyšla najevo malá komplikace a to ta, že po vypnutí efektu signál zůstal přeladěn. Proto je nutné přidat objekt *gate~*, který umožňuje úplný bypass efektu. Z *toggle* je vedena podmínka, že pokud je hodnota z *toggle* rozdílná od nuly (je tedy zapnutý), je vstupní signál nasměrován do druhého outletu, odkud pokračuje do efektu. Pokud je *toggle* vypnutý, je signál nasměrován do prvního outletu a pokračuje beze změny. Obdobně funguje spínání filtru, s tím rozdílem, že data ze senzoru po sepnutí ovládají mezní frekvenci a stejně tak reverb po sepnutí zvyšuje čas dozvuku.

3.3.9 Subpatch Rnd panning



Obr. 3.2.10: Detail subpatche Rnd panning

Zajímavým efektem pro granulární syntézu je náhodné rozložení velmi krátkých granulí do sterea, díky čemuž lze dosáhnout širšího prostorového vjemu. Pokud je *toogle* sepnut je signál náhodně posílán do levého a pravého výstupu z prvního a druhého outletu *objektu gate~*, pokud je *toggle* vypnutý, prochází signál třetím outletem do obou výstupů současně.

3.4 Praktické využití nástroje

Zvukově se syntezátor pohybuje mezi dvěma rovinami – klidovou a extrémní. Čím více jsou jednotlivé senzory osvětlovány, tím je zvuk zrnitější, frekvence vyšší, attack kratší a dochází ke spínání vybraných efektů. Výsledný zvuk granulární syntézy může znít kovově nebo naopak velmi jemně, a to v závislosti na nastavení obálky a delaye.

Základem ovládání nástroje je svícení do jednotlivých černých krystalů, na jejichž konci je vsazena fotodioda snímající změnu intenzity světla. Manipulace může na první pohled připomínat ovládání thereminu, jelikož intenzita světla se mění dle vzdálenosti zdroje od senzoru. Tak jako houslista hledá svůj vhodný smyčec či bubeník paličky, tak i k nástroji Ljós může použít jakýkoliv světelný zdroj. Může využít například i střídání dne a noci a zaznamenat tak 24hodinový cyklus změny intenzity světla do zvukové kompozice.

Syntezátor Ljós lze využít různými způsoby. V první řadě je využitelný pro audiovizuální live performance. Pro umělce je to interaktivní nástroj, který mu dává

možnost v reálném čase měnit zvuk díky kombinaci pohybu těla a intenzity světla. Hráč je tak schopen tvořit s jinou citlivostí než u tradičních kontrolerů.

Jak již bylo naznačeno, těžiště experimentálnosti nástroje spočívá v jeho interaktivním ovládání a předností je tak jeho vizuální efekt, který vzniká při manipulaci s ním. Upoutává diváckou pozornost a oživuje proces tvorby elektronické hudby. Syntezátor je však vhodný i do edukativní oblasti. Využitelný je pro studenty kompozice, audio inženýrství, muzikologie, estetiky, výtvarných umění a dalších oborů. Pro všechny zmíněné obory je důležité sledovat nové technologie a v nejlepším případě se je snažit zařadit do svých zažitých metod. V neposlední řadě můžeme uvažovat přínos nástroje i pro malé děti, kterým je tak umožněno se originálním způsobem přiblížit k procesu vzniku elektronického zvuku.

Závěr

V rámci bakalářské práce bylo zreflektováno historicko-kulturní pozadí multimediálního umění a interaktivních systémů. Počátky lze nalézt již v antické tragédii, barokní opeře nebo wagnerovském Gesamtkunstwerku. Ve 20. století pak impulzem ke vzniku netradičních nástrojů bylo hledání nových procesů kompozice, snaha o přesné vyjádření autorových myšlenek, hledání analogií mezi různými složkami umění či taktéž snaha o přesah a pronikání mezi těmito složkami. Na druhou stranu však interaktivní nástroje a systémy vznikaly i z důvodu problému absence interpreta v procesu elektronické či elektroakustické hudby. S interaktivitou mohl tak nástroj ožít a esteticky se přiblížit tradičnímu mechanickému nástroji.

Významnými konstruktéry a průkopníky 1. poloviny 20. století byli Lev Sergejevič Těrmen s nástrojem theremin, Jevgenij Murzin se syntezátorem ANS či Daphne Oram s technikou Oramics. Zásadní postavou pro vývoj počítačové hudby byl skladatel a architekt Iannis Xenakis, který v roce 1977 představil počítačový systém pro generování zvuku UPIC. Za pomoci tabletu připojeného k počítači mohl uživatel kreslit křivky a tím v reálném čase generovat a ovlivňovat spektrum výsledného zvuku. Na tento systém navazuje v současnosti spousta softwarů převádějících vizuální data na hudbu, mezi nimi např. IanniX nebo HighC.

Druhý polovina teoretické části shrnuje základní typy syntéz a představuje platformu Arduino a rozhraní Max/MSP. Konkrétněji jsou jednotlivé části rozebrány v kapitole Návrh.

Následující část pokrývá návrh syntezátoru. Vychází z navržených blokových schémat. Základním principem nástroje je proces přeměny intenzity osvětlení na zvukový signál. Intenzita světla je snímána za pomocí fotocitlivých diod a na nich odečtené hodnoty digitálně zpracovány v rámci platformy Arduino. Softwarová část nástroje je realizována v prostředí Max/MSP. Komunikace mezi Arduinem a Max/MSP probíhá za pomocí sériové linky. Na vstupu softwarového rozhraní syntezátoru tedy stojí sériová data, která jsou odečtena na analogových pinech desky Arduino.

Závěrečná kapitola se věnuje samotné realizaci syntezátoru. Tělo ovladače je zhotoveno za pomocí technologie 3D tisku. Nástroj disponuje deseti fotocitlivými senzory, které uživatel libovolně osvětluje. Tyto senzory ovládají následující parametry – frekvenci, tvorbu harmonického či neharmonického spektra, velikost zrna pro granulární

syntézu, stereo panning, delay, reverb, random pitch shifter, amplitudovou a modulační obálku. Manipulace s nástrojem je svým způsobem podobná ovládání thereminu – zvukový parametr je ovládán pohybem ruky, s tím rozdílem, že uživatel musí držet světelný zdroj.

K užívání nástroje není nutná znalost čtení tradičního notového zápisu či harmonie. Cílem je, spíš než melodii, tvořit barvu zvuku. Tím, že nástroj reaguje na změnu intenzity světla v reálném čase, může docházet k velmi nestabilním hodnotám. Proto si nástroj vyžaduje velmi soustředěné zacházení.

Výhodou realizace v prostředí Max/MSP a platformy Arduino je neustálá možnost rozšiřování. Jednou takovou možností by byla funkce přepínače, která by byla schopna vytvořit několik uživatelských vrstev a umožnit pohyb mezi nimi. Lze to demonstrovat na příkladu, kdy v první vrstvě by na senzory byly namapované stávající parametry, v další vrstvě by uživatel přepnul na podrobnější ovládání filtru, v další vrstvě by každý ze senzorů odpovídal jinému tónu apod.

Přijímaná sériová data lze převádět i na MIDI hodnoty, a tak by bylo možné ovládat externí hardware syntezátoru disponujícího MIDI připojením. Samotný ovladač lze použít i samostatně jako MIDI kontroler pro jiné DAW programy. K tomuto účelu je však potřeba vhodně přeprogramovat desku v IDE Arduino.

Syntezátor otevírá nové možnosti manipulace s nástrojem a vystupuje z konvenčního způsobu chápání tvorby zvuku. Podporuje tak svobodu tvůrčího myšlení.

Nástroj je určen především pro audiovizuální live performance. Upoutává diváckou pozornost a oživuje proces tvorby elektronické hudby. Syntezátor je však vhodný i pro edukativní oblasti. Využitelný je pro studenty kompozice, audio inženýrství, muzikologie, estetiky, výtvarných umění a dalších oborů.

Seznam použité literatury

[1] AGOSTINHO, G. a J. RATAJ. *Digitální technologie v hudební tvorbě pro akustické nástroje*. Praha: NAMU, 2016. ISBN 978-80-7331-419-4.
[2] FLAŠAR, M. *Hudba v kontextu multimédií*. In: Martin Flašar – Jana Horáková – Petr Macek a kol. Umění a nová média. Vyd. 1. Brno: Masarykova univerzita, 2011. ISBN 978-80-210-5639-8.
[3] FLAŠAR, M. *Poème électronique 1958. Le Corbusier – E. Varèse – I. Xenakis*, Brno: Masarykova univerzita, 2012. ISBN 978-80-210-5945-0.
[4] GUŠTAR, M. *Elektrofony: Historie, Principy, Souvislosti. Část II – elektronické nástroje*, Praha: 2008. ISBN 978-80-239-8447-7.
[5] LÉBL, V. *Způsob zápisu elektronické a konkrétní hudby*. In: Sborník přednášek o problémech elektronické hudby, Svazek I, Praha-Bratislava, Panton 1964, s. 33–50.
[6] NEJTEK, M. *Synestezie jako tvůrčí a perseční faktor u současných hudebně-scénických forem*, Brno: Janáčkova akademie múzických umění, 2015. ISBN 978-80-7460-081-4
[7] POMAJZLOVÁ, A. *Stín, druhá strana světla*. In: ZEMÁNEK, Jiří ed.: Ejhle světlo!, Brno: 2003. ISBN 80-86217-61-2. Dostupné také z: http://www.digitalniknihovna.cz/mzk/uuid/uuid:b907d0d0-0595-11e3-beb8-005056827e51.
[8] SELECKÝ, M. *Arduino: Uživatelská příručka*. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
[9] VODA, Z. a kol. *Průvodce světem Arduina.* 2. vyd. Bučovice: Nakladatelství Martin Stříž, 2017. ISBN 978-80-87106-93-8.

Internetové zdroje

[10] DRAJSAJTLOVÁ, K. *Světelný klavír v uměleckém díle Alexandera Nikolajeviče Skrjabina a Zdeňka Pešánka* [online]. Brno, 2012 [cit. 2017-12-03]. Dostupné z: <https://is.muni.cz/th/65026/ff_b/>.
[11] *The ANS synthesizer: Composing on a Photoelectronic Instrument* [online] [cit. 2017-12-3]. Dostupné z: http://www.theremin.ru/archive/ans.htm.
[12] *Oramics* [online] [cit. 2017-12-11]. Dostupné z: https://en.wikipedia.org/wiki/Oramics.
[13] *Marek Chołoniewski / Studio MCH* [online] [cit. 2017-11-24]. Dostupné z: http://www.studiomch.art.pl/index_ang.html.
[14] SUCHÁNEK, J. *Bio* [online] [cit. 2017-12-13]. Dostupné z: http://www.jiri-suchanek.net/en/bio/.
[15] DVOŘÁK, T. *Interaktivní práce* [online] [cit. 2017-12-13]. Dostupné z: http://www.floex.cz/interaktivni.php.

[16] FLAŠAR, M. *Interaktivní digitální nástroje pro live performance* [online] [cit. 2017-12-11]. Dostupné z: https://is.muni.cz/do/rect/el/estud/ff/ps15/eah/web/pages/11-live_performance.html.
[17] SYROVÝ, V. *Hudební signál a jeho syntéza*. In: Živá hudba 1986 [online] [cit. 2017-12-13]. Dostupné z: http://www.ziva-hudba.info/files/2014/04/140403201437_pdf_0.pdf.
[18] *Co je to Arduino?* [online] [cit. 2017-11-30]. Dostupné z: https://arduino.cz/co-je-to-arduino/.
[19] *Hardware & Related Initiatives: Arduino-compatible hardware, related initiatives, and other microcontroller platforms* [online] [cit. 2017-12-5]. Dostupné z: http://playground.arduino.cc/main/similarBoards.
[20] *Oficiální dokumentace Max 7* [online] [cit. 2017-12-11]. Dostupné z: https://docs.cycling74.com/max7/.
[21] *Max (software)* [online] [cit. 2017-12-11]. Dostupné z: https://en.wikipedia.org/wiki/Max_(software).
[22] *OpenMusic* [online] [cit. 2018-08-12]. Dostupné z: https://en.wikipedia.org/wiki/OpenMusic.
[23] *Integra Live* [online] [cit. 2018-08-12]. Dostupné z: http://integra.io/integralive/.
[24] *AudioMulch* [online] [cit. 2018-08-12]. Dostupné z: https://en.wikipedia.org/wiki/AudioMulchinte.
[25] *Fotodioda* [online] [cit. 2017-12-9]. Dostupné z: https://eluc.kr-olomoucky.cz/verejne/lekce/61.
[26] *Oficiální dokumentace Arduino Mega 2560 Rev3* [online] [cit. 2017-12-11]. Dostupné z: https://store.arduino.cc/arduino-mega-2560-rev3.

[27] *Studijní materiály předmětu Tvorba umělého zvuku, jeho zpracování a řízení*.

Zdroje obrázků

[A] *Obr. 5: Schéma desky Arduio Mega 2560* – vygenerován pomocí programu Fritzing.org

Seznam použitých zkratek a veličin

ASCII – American Standard Code for Information Interchange (americký standardní kód pro výměnu informací)
CEMAMu – Centre d'Etudes de Mathématique et Automatique Musicales
DAW – Digital Audio Workstation
EAH – elektroakustická hudba
EEG – elektroencefalografie
EEPROM – Electrically Erasable Programmable Read-Only Memory (elektronicky vymazatelná paměť pouze pro čtení)
IDE – Integrated Development Environment (integrované vývojové prostředí)
IRCAM – Institut de recherche et coordination acoustique/musique
LFO – Low Frequency Oscillator (nízkofrekvenční oscilátor)
OSH – Open Source Hardware (otevřený hardware)
OSS – Open Source Software (otevřený software)
Pd – Pure Data
PWM – Pulse Wide Modulation (modulace šířky pulsů)
UART – Universal Asynchronous Receiver-Transmitter (univerzální asynchornní přijímač/vysílač)
USB – Universal Serial Bus (univerzální sériová sběrnice)

t_A .... doba trvání parametru attack
t_D .... doba trvání parametru decay
t_S .... doba trvání parametru sustain
t_R .... doba trvání parametru release.
t_{DSR} .... dobu součtu trvání decay, sustain a release
t_{ADSR} .... doba celkového trvání obálky

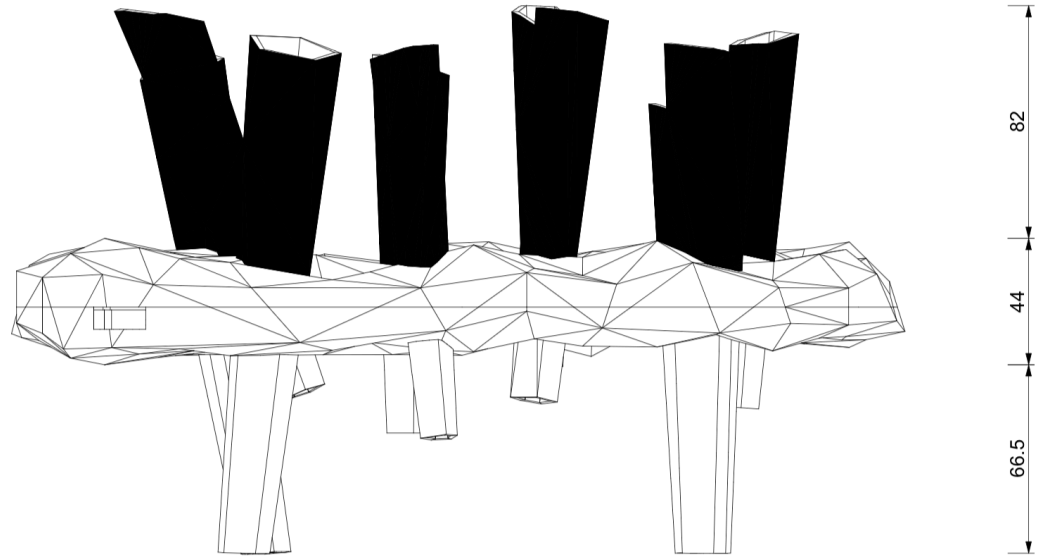
Přílohy

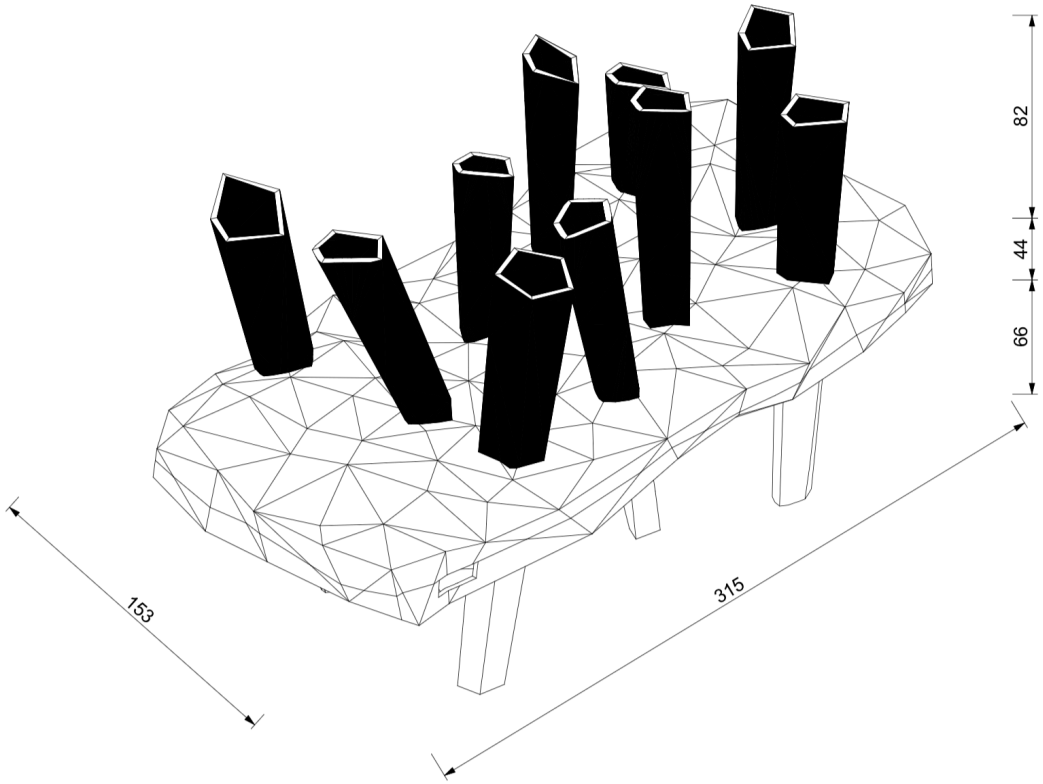
A Program pro čtení hodnot z analogových pinů Arduina

```
void setup()
{
 Serial.begin(9600);
}

int smooth(int pinNo)                    //funkce smoothing
analogových vstupů
{
 int i;
 int value = 0;
 int numReadings = 5;

 for (i = 0; i < numReadings; i++)
 {
  value = value + analogRead(pinNo);    //čtení dat z analogového pinu
  delay(1);                    // 1ms zpoždění pro větší stabilitu mezi jednotlivými čteními
 }

 value = value / numReadings;          // průměr ze všech čtení

 return value;
}

void loop() {

 /*int tick = millis();*/              // měření prodlevy mezi měřeními -       začátek řady

 int i;
 for (i = 0; i < 10; i++)
 {
  if (i < 9)                    // tisk zprůměrované
```

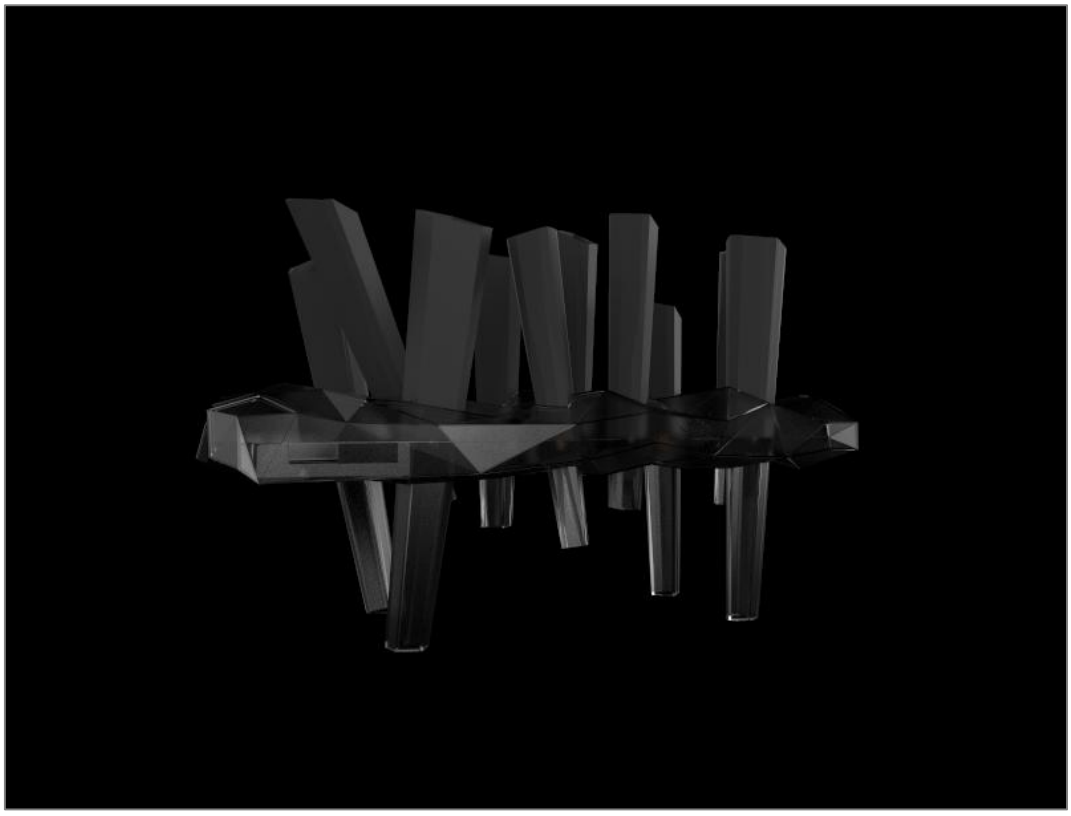
```
naměřené hodnoty z pinu 0–8 do sériového monitoru
  {
    Serial.print(smooth(i));
    Serial.print(" ");              // oddělení jednotlivých hodnot mezerou
  }
  else
  {
   Serial.println(smooth(i));       // tisk zprůměrované naměřené hodnoty na
posledním pinu 9 a oddělení řádku v sériovém monitoru
  }
  }
 /*int tock = millis();
 Serial.println(tock - tick);*/       // měření prodlevy mezi měřeními - konec řady

 delay(50);                    // 50ms pauza mezi opakováním void(loop)


}
```

B Návrh a rozměry těla ovladače





C Render 3D modelu ze softwaru Rhinocerus



D Prototyp Ljós