

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mitja Cerkvénik

# Sintetizator zvoka in sekvencer bobnov z MIDI vmesnikom

DIPLOMSKO DELO  
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2012



Št. naloge: 01827/2012

Datum: 02.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MITJA CERKVENIK**

Naslov: **SINTETIZATOR ZVOKA IN SEKVENCER BOBNOV Z MIDI  
VMESNIKOM**  
**SOUND SYNTHESIZER AND DRUM SEQUENCER WITH MIDI  
INTERFACE**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Glasbeni inštrumenti, ki oblikujejo zvok s pomočjo elektronike, so danes zelo razširjeni. Za njihovo komuniciranje z računalniki je bilo razvitih veliko orodij, med katerimi je posebno razširjen standard MIDI. Razvijte in izdelajte glasbeni inštrument, ki vključuje sintetizator in sekvencer bobnov z MIDI vmesnikom. Inštrument naj bo zasnovan tako, da bo primeren za samostojno nastopanje. Omogoča naj ustvarjanje in predvajanje tako ritma kot melodije. Izberite ustrezne mehanske, elektronske in računalniške komponente ter izdelajte potrebno programsko opremo. Pravilnost delovanja preverite in podajte možne načine za izboljšave.

Mentor:

prof. dr. Dušan Kodek

Dekan:

prof. dr. Nikolaj Zimic



# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani      Mitja Cerkvenik,  
z vpisno številko      63010018,  
sem avtor diplomskega dela z naslovom:

Sintetizator zvoka in sekvencer bobnov z MIDI vmesnikom

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Dušana Kodeka
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 27.6.2012

Podpis avtorja:

# **Zahvala**

Zahvaljujem se mentorju za pomoč pri izdelavi diplomske naloge.

Prav tako gre posebna zahvala staršem, ker so me podpirali v času študija.

Posebna zahvala gre tudi Darku Kendi in Zoranu Kendi ter podjetju Augmentech d.o.o., za številne nasvete, pomoč, nudenje prostora in opreme za realizacijo inštrumenta, ki je rezultat tega diplomskega dela.

Diplomsko delo posvečam svojim staršem.

# Kazalo

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Povzetek .....                                                     | 1  |
| Abstract.....                                                      | 2  |
| 1. Uvod.....                                                       | 3  |
| 2. Sintetizator zvoka z MIDI priključkom .....                     | 5  |
| 2.1. Vezje.....                                                    | 5  |
| 2.1.1. Arduino Uno .....                                           | 5  |
| 2.1.2. Vezje MIDI priključka .....                                 | 6  |
| 2.1.3. Ostali elementi.....                                        | 7  |
| 2.2. Program.....                                                  | 8  |
| 2.2.1. Pulse Width Modulation (PWM) in nastavitve časovnikov ..... | 8  |
| 2.2.2. Nastavitve registrov.....                                   | 11 |
| 2.2.3. Direct digital synthesis in granularna sinteza.....         | 12 |
| 2.2.4. Kromatična tabela inkrementov.....                          | 14 |
| 2.2.5. Prekinitveni servisni program.....                          | 15 |
| 3. Sekvencer in sintetizator bobnov .....                          | 17 |
| 3.1. Vezje.....                                                    | 17 |
| 3.1.1. Arduino Mega 2560 .....                                     | 17 |
| 3.1.2. Ostali elementi.....                                        | 18 |
| 3.2. Program.....                                                  | 19 |
| 4. Upravljanje inštrumenta.....                                    | 26 |
| 4.1. Upravljanje sintetizatorja.....                               | 27 |
| 4.2. Upravljanje sekvencerja.....                                  | 27 |
| 5. Sklepne ugotovitve .....                                        | 28 |
| Priloge .....                                                      | 30 |

|                                 |    |
|---------------------------------|----|
| 1. Tabela notnih frekvenc ..... | 30 |
| Seznam slik.....                | 31 |
| Literatura.....                 | 32 |

# Seznam uporabljenih kratic in simbolov

|                    |                                                        |
|--------------------|--------------------------------------------------------|
| PWM                | (angl.) Pulse Width Modulation                         |
| D/A                | Digitalno/Analogno                                     |
| MIDI               | (angl.) Musical Instrument Digital Interface           |
| LED                | (angl.) Light Emitting Diode                           |
| DIN                | (nem.) Deutsches Institut für Normung                  |
| RX                 | (angl.) Receive, sprejemnik                            |
| OCnx               | (angl.) Output Compare output                          |
| OCRnx              | (angl.) Output Compare Register                        |
| TCNTx              | (angl.) Timer/Counter                                  |
| TCCRnx             | (angl.) Timer/Counter Control Register                 |
| WGM                | (angl.) Waveform Generation Mode                       |
| COMnx              | (angl.) Compare Output Mode                            |
| CSxx               | (angl.) Clock Select                                   |
| TOIE <sub>n</sub>  | (angl.) Timer/Counter n Overflow Interrupt Enable      |
| TIMSK <sub>n</sub> | (angl.) Timer/Counter n Interrupt Mask Register        |
| FOCnx              | (angl.) Force Output Compare                           |
| W                  | (angl.) Write, pisanje                                 |
| R                  | (angl.) Read, branje                                   |
| R/W                | (angl.) Read/Write, branje/pisanje                     |
| DDS                | (angl.) Direct Digital Synthesis                       |
| SRAM               | (angl.) Static Random-Access Memory                    |
| PSP                | Prekinitveni servisni program                          |
| DDRx               | (angl.) Data Direction Register                        |
| ICES <sub>n</sub>  | (angl.) Input Capture Edge Select                      |
| ICNC <sub>n</sub>  | (angl.) Input Capture Noise Canceler                   |
| ICIE <sub>n</sub>  | (angl.) Timer/Counter n Input Capture Interrupt Enable |
| OCIE <sub>nx</sub> | (angl.) Output Compare Interrupt Enable                |



## **Povzetek**

Elektronska glasba se je zelo uveljavila v zadnjih desetletjih. Ustvarja se predvsem z računalniškimi programi, ker so bolj dostopni od sintetizatorjev. Uporabniški vmesnik glasbenih aplikacij ne omogoča tako hitrega upravljanja inštrumenta kot strojna inačica. Pogosta rešitev je uporaba MIDI krmilnika. To je strojni uporabniški vmesnik, s katerim lahko upravljamo zvoke na računalniku, povezan preko MIDI kabla.

To diplomsko delo je posvečeno sintetiziranju zvokov in ritmičnemu urejanju teh zvokov v zaporedje preko strojnega uporabniškega vmesnika. Zasnovan in izdelan je bil sintetizator z MIDI vhodom in sekvencerjem. Melodične zvoke je možno igrati in oblikovati v realnem času preko MIDI vhoda in potenciometrov. Ritmične zvoke je prav tako možno oblikovati preko potenciometrov. Sekvencer je del sistema, ki skrbi za vnos in predvajanje zvokov v določenem ritmu. Sestavljen je iz stikal, preko katerih se vnaša ritem, in LED diod, ki ga prikazujejo.

Cilj diplomske naloge je bil raziskati možnosti sintetiziranja zvoka. Ta cilj je bil dosežen z uporabo razvojne ploščice Arduino. Ta nudi PWM izhod, ki ga z ustreznim nizko prepustnim filtrom, lahko uporabimo kot D/A pretvornik. Na koncu je nastal polno delujoč prototip sintetizatorja z MIDI vhodom in sekvencerjem.

## **Ključne besede:**

Sintetizator, MIDI, sekvencer, Arduino, PWM, granularna sinteza

## Abstract

Electronic music has gained popularity in the last decades. Most of it is produced with computer software applications because they are more accessible than synthesizers. The user interface of music applications does not allow such rapid instrument management as the hardware version. A common solution is the use of a MIDI controller which is a physical user interface that can manage sounds on the computer connected with a MIDI cable.

In this thesis we are focusing on sound synthesis and sequencing of sounds in a rhythm through a physical user interface. A synthesizer with a MIDI input and a sequencer was designed and built. Melodies can be played through the MIDI input and the sound can be modulated through potentiometers. Rhythmic sounds can also be manipulated through potentiometers. A sequencer is the part of the system which handles the input of a rhythmic pattern and its playback. It consists of switches through which the pattern is imported and LEDs that show the entered rhythm.

The aim of this work was to explore the possibility of sound synthesis. This goal was achieved using the Arduino development board. It provides a PWM output which in conjunction with a suitable low pass filter can be used as a D/A converter. In the end a fully functional synthesizer prototype with a MIDI input and a sequencer was built.

## Key words:

Synthesizer, MIDI, sequencer, Arduino, PWM, granular synthesis

# 1. Uvod

Leta 2013 bo minilo 100 let, odkar je Luigi Russolo napisal futuristični manifest »Umetnost hrupa«. V njem je, med drugim, spodbujal k ustvarjanju novih glasbenih inštrumentov s pomočjo elektronike in drugih, tedaj novih tehnologij. Elektronski inštrumenti so v tistem času že obstajali, vendar so elektronske komponente, ki jih sestavljajo, prišle v masovno uporabo šele v 30. letih prejšnjega stoletja. Štirideset let za tem so nastali prvi digitalni sintetizatorji in začeli postopoma nadomeščati analogne. Njihova prednost, v primerjavi z analognimi, leži predvsem v vzdržljivosti, v bistveno manjši občutljivosti na temperaturo in staranje ter v možnosti shranjevanja nastavitvev. Kljub temu analogni sintetizatorji danes uživajo nekakšen kulturni status zaradi »toplejšega« zvoka in tudi zaradi njihove redkosti. Prvi digitalni metodi sintetiziranja zvoka sta bili frekvenčna in fazna modulacija. Kasneje so se uporabljali vzorčeni zvoki in druge tehnike, kot na primer granularna sinteza. Novejši algoritmi za generiranje digitalno sintetiziranega zvoka so velikokrat napisani tako, da bi generiran zvok oponašal tistega analognih sintetizatorjev. Zadnja generacija digitalnih sintetizatorjev pa uporablja digitalno procesiranje signalov in programske algoritme za simulacijo elektronskih vezij analognih sintetizatorjev. Očitno je, da se analogni in digitalni svet v sodobni glasbi prepletata na več nivojih.

Zanesljivost in natančnost sta ključni lastnosti, zaradi katerih je digitalna tehnologija prevladala nad analogno. To še posebej velja pri napravah, od katerih je odvisno življenje človeka. Glasbila niso tak primer, vendar je njihova zanesljivost zelo pomembna. Nezanljiva glasbila bi lahko brez premislekov opredelili kot neuporabna. Natančnost inštrumenta pa je včasih lahko del umetniškega izražanja. Uporabiti razglašen klavir za izvedbo določene skladbe, na primer, je lahko izbira umetnika. Digitalna tehnologija je omogočila zelo »čist«, oster zvok, z izračunljivimi omejitvami, brez šuma, ki je tako rekoč okarakteriziral analogne sintetizatorje. Cenovna dosegljivost in zmogljivost digitalne glasbene opreme naraščata. Kljub temu se pogosto sliši zvoke slabše kvalitete, tako v komercialnih radijskih skladbah kot v bolj sofisticirani glasbi. Razlog za to je ekspresivnost. Običajno se kvaliteto zvoka poslabša tako, da se zvok posname skozi filtre, ki omejijo frekvenčni spekter ali znižajo bitno resolucijo in vzorčevalno frekvenco.

Večina sodobne glasbe je ustvarjena z računalniškimi glasbenimi programi. Ti so se v zadnjih dveh desetletjih zelo razširili in razvili. Tudi v živih nastopih se vedno bolj pogosto

vidijo prenosni računalniki na odru. Vnašati ritme s kliki na miško je zamudno, zato so v teh primerih ritmi že posneti. Možnosti za improvizacijo ni veliko. Za to so veliko primernejše t. i. »ritem mašine«<sup>1</sup> ali samplerji<sup>2</sup> s sekvencerjem. Ritem mašina je elektronski glasbeni inštrument, ki predvaja zvoke bobnov. Ti so bodisi vzorčeni bodisi sintetizirani s pomočjo analognih ali digitalnih elektronskih komponent. Običajno ritem mašina vsebuje potenciometre, ki omogočajo spreminjanje zvokov elektronskih bobnov v realnem času. Zvoke se ureja v ritem s sekvencerjem. To je del ritem mašine, ki določa zaporedje zvokov v času. Za vnos ritma ima 16 stikal, ki predstavljajo šestnajstinke v taktu. Vključena stikala določijo, da se izbrani zvok sproži na dobo, ki jo predstavljajo v taktu. Zaporedje šestnajstink se ponavlja v neskončni zanki. Obstajajo sekvencerji, ki so fizično ločeni od sintetizatorja. Sampler je prav tako elektronski glasbeni inštrument. Predvaja zgolj vzorčene zvoke in omogoča njihovo manipulacijo s potenciometri. Nekateri samplerji vsebujejo sekvencer, klaviaturo ali posebne tipke oblečene v blazinice, po katerih se udari, da se sproži zvok.

Mikrokontrolniki so cenovno dostopni in dovolj zmogljivi, da je mogoče na njih zgraditi »low-fidelity« inštrumente. To so elektronski inštrumenti, ki imajo bitno resolucijo nižjo od 16 bitov in vzorčevalno frekvenco pod 44100 Hz. V diplomskem delu je uporabljeno razvojno okolje Arduino. Zmogljivost tega mikrokontrolnika je zadostna za 8 bitni izhodni signal. Ideja je bila izdelati sintetizator zvoka z MIDI priključkom in z možnostjo oblikovanja signala v realnem času ter sekvencer za programiranje ritma sintetiziranih bobnov (t.i. ritem mašina). Zvok sintetizatorja je možno oblikovati v realnem času s pomočjo potenciometra. Možna je izbira tipa osnovnega signala kot žagastega, pravokotnega, sinusnega ali kot šum. Višino tona se določi z MIDI klaviaturo povezano na MIDI priključek sintetizatorja. Polifonija je 3, to pomeni, da lahko igramo tri tone hkrati, kar zadošča za igranje akordov. Ritem bobnov se vnaša s 16 stikali. Sintetiziran zvok bobnov je možno oblikovati (ton, trajanje). S potenciometri je določena glasnost izhodnega signala.

---

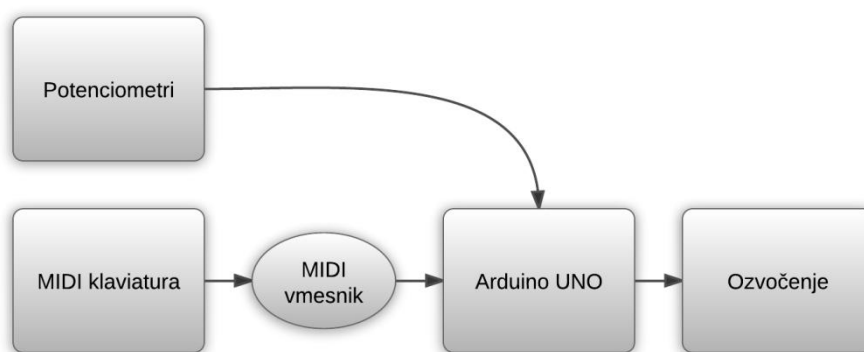
<sup>1</sup> Izraz »ritem mašina«, ki se nanaša na sintetizator zvokov bobnov in sekvencer, nima uveljavljene slovenske različice. Strokovnjaki uporabljajo za ta inštrument skoraj izključno ta pogovorni izraz. Nekateri so predlagali strokovno različico »elektronski sintetizatorski kompozitor ritmov« ali »generator ritmičnih zvokov«. Da ne bi povzročali dodatne zmede, bomo v tem diplomskem delu uporabili izraz »ritem mašina«.

<sup>2</sup> Sampler je angleški izraz, ki nima slovenske različice, ko gre za tukaj omenjeno napravo. Sicer bi izraz lahko prevedli kot »vzorčevalnik«.

## 2. Sintetizator zvoka z MIDI priključkom

### 2.1. Vezje

Osrednji del sintetizatorja predstavlja razvojna ploščica Arduino Uno. Je zelo razširjena, na voljo je veliko knjižnic in podpora s strani skupnosti Arduino. DIN konektor je standard za MIDI in je povezan na Arduino preko optospojnika. Na konektor lahko povežemo poljubno klaviaturo z MIDI izhodom. Drsni in krožni potenciometri služijo za določanje glasnosti izhodnega signala in njegove modulacije. Blok shema (Slika 1) prikazuje, kako so sestavni deli povezani v celoto.



Slika 1: Blok shema sintetizatorja z MIDI vmesnikom

#### 2.1.1. Arduino Uno

Arduino je odprtokodna platforma. Razvita je bila, da bi se olajšalo interdisciplinarno uporabo elektronike. Napravo sestavlja ena sama ploščica (Slika 2) z Atmel-ovim procesorjem in z digitalnimi ter analognimi vhodi oziroma izhodi [1].

Arduino Uno je osnovan na mikrokrmilniku ATmega328. Ima 14 digitalnih vhodov oziroma izhodov in 6 analognih vhodov. 6 digitalnih izhodov lahko uporabljamo kot Pulse Width Modulation (PWM) izhode. Procesor deluje s frekvenco 16MHz. Napaja se preko USB ali preko napajalnega vtiča. Program naložimo prav tako preko USB vtiča, pišemo pa ga v prirejenem programskem jeziku, ki je podoben programskemu jeziku C.

Na ploščico Arduino so priključeni potenciometer, 6.35 mm mono avdio priključek in enostavno vezje za galvansko ločevanje DIN priključka.



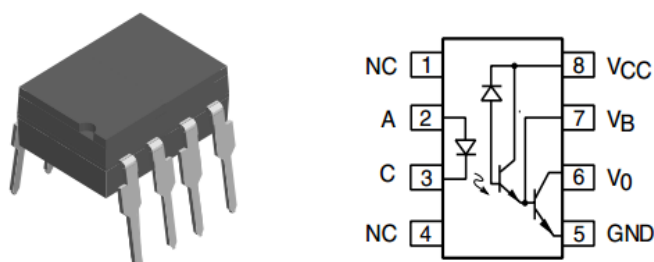
Slika 2: Arduino UNO razvojna ploščica

### 2.1.2. Vezje MIDI priključka

Povezava med Arduino in MIDI napravo zahteva izdelavo enostavnega vezja. MIDI protokol uporablja serijsko komunikacijo, ki poteka preko tokovne zanke. Za prenos digitalnega signala se uporablja tok 5mA za logično ničlo in odsotnost toka za logično enko [2]. Tokovni signal je potrebno spremeniti v napetostnega zato, da ga lahko Arduino prebere.

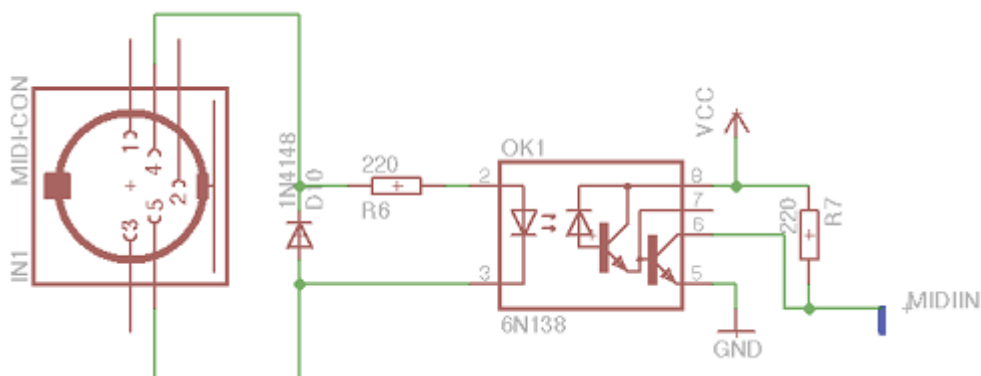
MIDI specifikacije določajo, da se nožici 4 in 5 DIN konektorja uporabita za prenos MIDI sporočil. Masa MIDI kabla mora biti priključena izključno na nožici 2 na MIDI izhodu. V primeru, da sta masa vhoda in izhoda povezani, lahko nastane šum v izhodnem signalu. Razlog za to je morebitna razlika v napetosti med izhodom in vhodom. Pomembno je, da sta ploščica Arduino in MIDI klaviatura galvanjsko ločeni. To dosežemo z uporabo optospojnika. Poleg tega, potrebujemo za realizacijo vezja še DIN konektor, diodo in dva  $220\Omega$  upora.

Optospojnik 6N138 (Slika 3) je v ohišju z 8 nožicami. Iz specifikacij [3] je razvidno, katere nožice moramo priključiti.



Slika 3: Optospojnik 6N138

Nožico 4 DIN konektorja povežemo na nožico 2 optospojnika preko  $220\Omega$  upora, nožico 5 pa neposredno na nožico 3 optospojnika. Med tema nožicama DIN konektorja je priporočljivo povezati še diodo, tako kot prikazuje shema vezja (Slika 4). Dioda deluje kot varovalka za LED diodo v optospojniku v primeru, da bi tok tekel v nasprotno smer. Vezje deluje tudi brez diode. Maso povežemo na nožico 5, napajanje +5V pa na nožico 8.  $V_0$  in dvizni upor priklopimo na Arduinov serijski vhod RX, to je digitalni vhod 0.

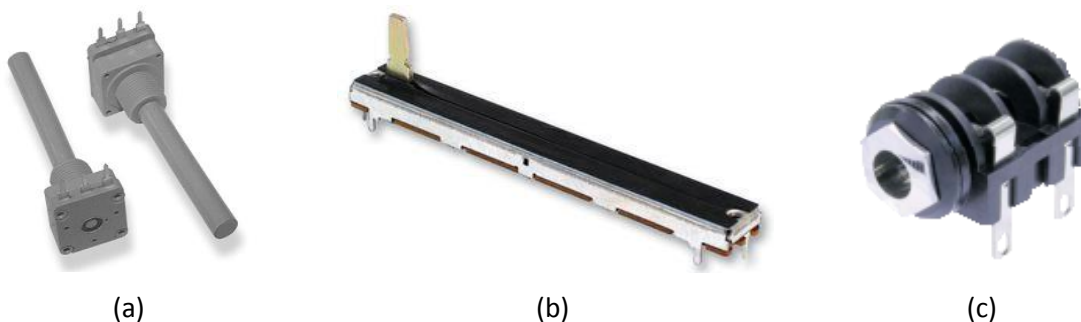


Slika 4: Shema vezja za MIDI priključek

### 2.1.3. Ostali elementi

Krožna linearna potenciometra (Slika 5.a) z upornostjo  $40\text{ k}\Omega$  sta povezana na maso (začetek potenciometra) in na +5V (konec potenciometra). Drsnik posameznega potenciometra pa je priključen na analogni vhod A0 oziroma A1 na Arduino plošči.

Drsni linearni potenciometer (Slika 5.b) z upornostjo  $10\text{ k}\Omega$  je povezan na Arduinov digitalni izhod 3 (konec potenciometra) in na maso (začetek potenciometra). Nožica drsnika pa je povezana na 6.35 mm mono avdio priključek (Slika 5.c). Potenciometer določa znižanje nivoja glasnosti izhodnega signala.



Slika 5: Krožna potenciometra (a), drsnni potenciometer (b) in 6.35 mm mono avdio priključek (c)

## 2.2. Program

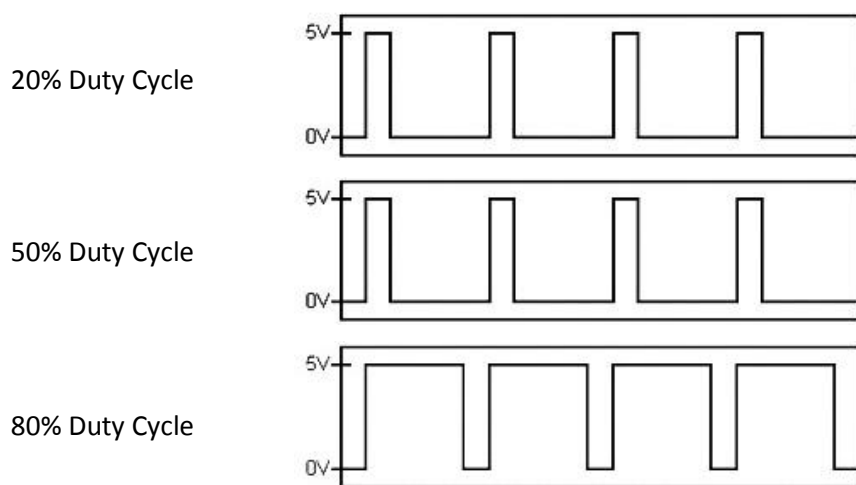
Programski del je dejansko jedro sistema. Program napišemo s posebnim razvojnim orodjem. Omogoča nam uvoz knjižnic, prevajanje kode in njeno nalaganje na Arduino ploščico. Trenutno je na voljo različica 1.0, s katero smo program razvili.

Z ukazom *include* se uvozi želene knjižnice. Za sintetizator zvoka potrebujemo knjižnice za upravljanje vhodov in izhodov, prekinitve, programski spomin (Flash memory) ter MIDI sporočila. Nato se deklarira globalne spremenljivke, konstante in morebitne tabele.

Po vklopu ploščice Arduino se zažene funkcija *setup()*. V tej funkciji najprej zaženemo *MIDI.begin()*, da vzpostavimo zahtevane pogoje MIDI [4]. S funkcijo *MIDI.turnThruOff()* izklopimo posredovanje vhodnega MIDI signala na izhod, saj ga nismo implementirali. Nato postavimo potrebne zastavice v registrih tako, da omogočimo *PWM* in sproženje prekinitev ter določimo urin delilnik enak 1. Na koncu inicializiramo vhode oziroma izhode, ki jih bomo uporabljali. Funkciji *setup()* sledi funkcija *loop()*, ki se izvaja v neskončni zanki. Tu se čaka na MIDI sporočila in se posodablja vrednosti potenciometrov. S frekvenco 31372,55 Hz se sproži prekinitev. V prekinitvenem servisnem programu se izračuna izhodno vrednost. V nadaljevanju je program podrobneje opisan.

### 2.2.1. Pulse Width Modulation (PWM) in nastavitve časovnikov

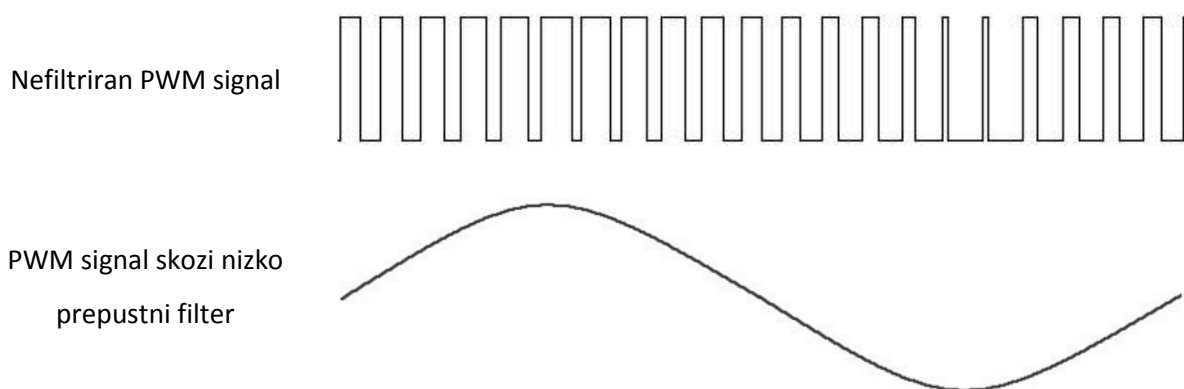
Ploščica Arduino nima pravega analognega izhoda. Ima pa implementirano možnost uporabe Pulse Width Modulation na določenih digitalnih izhodih.



Slika 6: Duty Cycle pri PWM

Preklopi med nizkim (0V) in visokim (+5V) stanjem ustvarijo pravokotni signal (Slika 6). Razmerje med časom ko je signal visok in časom trajanja periode, imenujemo *Duty Cycle* in določa povprečno vmesno vrednost (med 0V in 5V).

PWM izhod s primernim nizko prepustnim filtrom se lahko uporabi kot D/A pretvornik. Na spodnji sliki (Slika 7) lahko primerjamo PWM signal brez filtra in isti signal filtriran z nizko prepustnim filtrom.



**Slika 7: Nefiltriran in filtriran PWM signal**

Dolžina periode PWM signala je odvisna od časovnikov mikrokrmilnika in od sistemske ure. Arduino Uno ima tri notranje časovnike. Vsak ima dva primerjalna registra (OCR $n$ A in OCR $n$ B, kjer je  $n$  številka časovnika) s katerima se določi *Duty Cycle* na skupno šest različnih PWM izhodih. Časovniki imajo različne načine delovanja. Obstajata načina *Fast PWM* in *Phase-Correct PWM*.

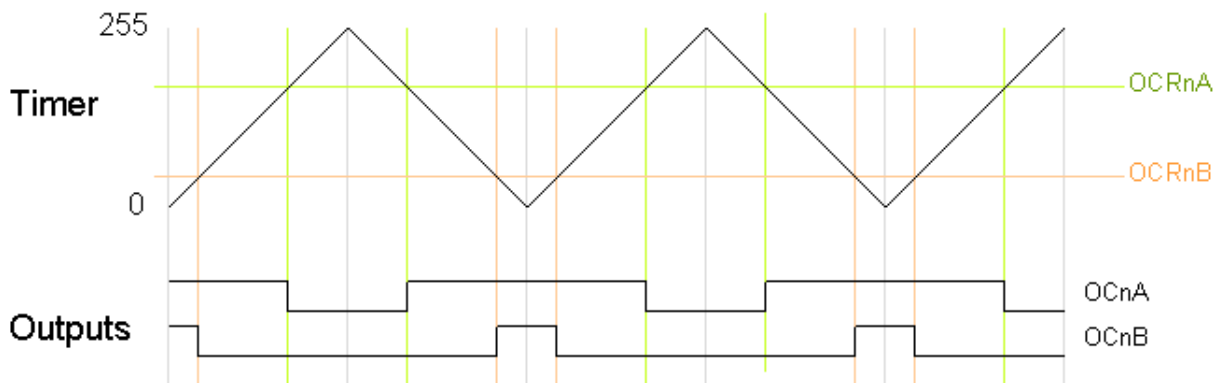
| Izhod časovnika | Arduinov pin |
|-----------------|--------------|
| OC0A            | 6            |
| OC0B            | 5            |
| OC1A            | 9            |
| OC1B            | 10           |
| OC2A            | 11           |
| OC2B            | 3            |

**Tabela 1: Razpored primerjalnih registrov po PWM izhodih**

Pri *Fast PWM* 8 bitni časovnik teče od 0 do 255, nato začne ponovno od 0 naprej. Perioda tega časovnika traja torej 256 urinih period. 8 bitni primerjalni register ima najvišjo vrednost

255 in *Duty Cycle* je določen kot  $(N+1)/256$ , kjer je  $N$  vrednost primerjalnega registra. To pomeni, da nimamo na voljo *Duty Cycle* 0%, ampak je najnižja vrednost  $1/256 = 0.004\%$ .

*Phase-Correct PWM* uporablja časovnik tako, da šteje od 0 do 255 in nazaj do 0 (Slika 8). Izhod se postavi v nizko stanje, ko je vrednost naraščajočega časovnika enaka tisti v primerjalnemu registru, in se ponovno postavi v visoko stanje, ko se vrednosti ujemata med odštevanjem časovnika. Na tak način lahko dosežemo *Duty Cycle* od 0% do 100%.



Slika 8: Phase-Correct PWM

V primerjavi s Fast PWM, Phase-Correct PWM razpolovi sistemsko uro zaradi časovnika, ki šteje v obe smeri. PWM cikel se torej ponavlja s frekvenco  $F_s$ .

$$F_s = \frac{16\text{MHz}}{255 * 2} = 31372,55 \text{ Hz}$$

Za izmenični signal potrebujemo vsaj dve vrednosti. To pomeni, da je najvišja dosegljiva frekvenca takega sistema

$$\frac{31372,55}{2} = 15686,28 \text{ Hz}$$

Arduinov programski jezik omogoča enostavno uporabo PWM. Načeloma je dovolj ukaz `analogWrite(pin, dutyCycle)`, kjer je `pin` eden od digitalnih izhodov s PWM in `dutyCycle` je vrednost med 0 in 255. Problem pri ukazu `analogWrite` je, da ne ponuja nobene možnosti nadzora nad frekvenco, s katero naj bi se vrednosti zapisale na izhod. Poleg tega pa je nemogoče izkoristiti mikroprocesor med zaporednimi zapisi, ker bi povzročili »luknje« oziroma prekinitve v izhodnem signalu. Neposredna manipulacija registrov je hitrejša od ukaza `analogWrite()`. Izhodno 8 bitno vrednost moramo zapisati v register OCRnB na redne intervale. To dosežemo tako, da se ob vsakem prelivu časovnika sproži prekinitveni servisni program.

### 2.2.2. Nastavitev registrov

Iz tabele izhodov časovnikov (Tabela 1) lahko razberemo, da moramo za uporabo PWM na nožici 3 uporabiti časovnik 2 (TCNT2 – Timer/Counter). Z ukazom `_BV()` določimo kateri bit v registru TCCR2x (Timer/Counter Control Register 2x) se bo postavil v visoko stanje.

Postavljen bit WGM20 (Waveform Generation Mode) v TCCR2A določi način delovanja časovnika kot PWM Phase-Correct. Bit COM2B1 določi, da se izhod OC2B postavi v nizko stanje, ko se naraščajoč časovnik ujema s primerjalnim registrom in gre v visoko stanje, ko se ujema pri odštevanju časovnika.

V registru TCCR2B postavljen bit CS20 določi, da je urin delilnik 1.

Bit TOIE2 (Timer/Counter2 Overflow Interrupt Enable) v registru TIMSK2 (Timer/Counter2 Interrupt Mask Register) določa ali se bo sprožila prekinitev, ko bo prišlo do preliva v časovniku TCNT2.

**Register TCCR2A**

| Bit              | 7      | 6      | 5      | 4      | 3 | 2 | 1     | 0     |
|------------------|--------|--------|--------|--------|---|---|-------|-------|
| (0xB0)           | COM2A1 | COM2A0 | COM2B1 | COM2B0 | - | - | WGM21 | WGM20 |
| Read/Write       | R/W    | R/W    | R/W    | R/W    | R | R | R/W   | R/W   |
| Začetna vrednost | 0      | 0      | 0      | 0      | 0 | 0 | 0     | 0     |

**Register TCCR2B**

| Bit              | 7     | 6     | 5 | 4 | 3     | 2    | 1    | 0    |
|------------------|-------|-------|---|---|-------|------|------|------|
| (0xB1)           | FOC2A | FOC2B | - | - | WGM22 | CS22 | CS21 | CS20 |
| Read/Write       | W     | W     | R | R | R/W   | R/W  | R/W  | R/W  |
| Začetna vrednost | 0     | 0     | 0 | 0 | 0     | 0    | 0    | 0    |

**Register TIMSK2**

| Bit              | 7 | 6 | 5 | 4 | 3 | 2      | 1      | 0     |
|------------------|---|---|---|---|---|--------|--------|-------|
| (0x70)           | - | - | - | - | - | OCIE2B | OCIE2A | TOIE2 |
| Read/Write       | R | R | R | R | R | R/W    | R/W    | R/W   |
| Začetna vrednost | 0 | 0 | 0 | 0 | 0 | 0      | 0      | 0     |

Koda za nastavitev registrov in določanje digitalnega izhoda 3 kot izhod (PWM\_PIN je definiran kot 3):

```
TCCR2A = _BV(COM2B1) | _BV(WGM20);
TCCR2B = _BV(CS20);
TIMSK2 = _BV(TOIE2);
pinMode(PWM_PIN, OUTPUT);
```

### 2.2.3. Direct digital synthesis in granularna sinteza

Direct Digital Synthesis (DDS) je digitalna metoda za generiranje signalov poljubnih frekvenc in oblik na podlagi višje frekvence referenčne ure [5]. Zaradi digitalne narave algoritma DDS omogoča hitre spremembe med izhodnimi frekvencami in poljubno visoko frekvenčno resolucijo. Pri programski implementaciji DDS algoritma potrebujemo:

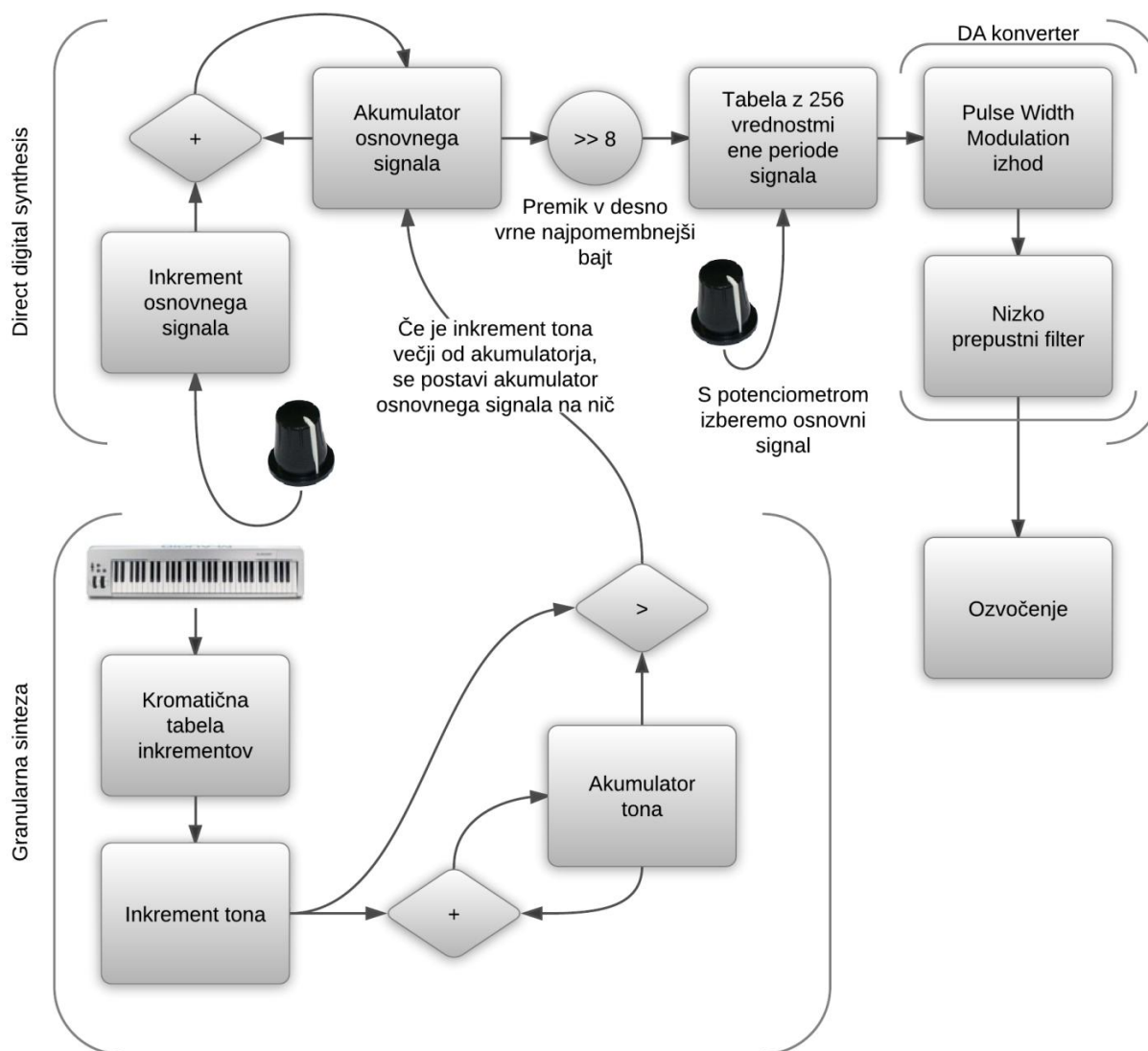
- *Akumulator*: spremenljivka tipa 16 bitno nepredznačeno celo število
- *Inkrement*: spremenljivka tipa 16 bitno nepredznačeno celo število
- *Tabelo* z zaporednimi vrednostmi ene periode želenega signala shranjene kot konstante
- *Digitalno analogni pretvornik*: priskrbljen z enoto PWM
- *Referenčno uro*: ura pridobljena iz strojnega časovnika v mikroprocesorju *ATmega*

Akumulatorju seštevamo *inkrement*. Najpomembnejši bajt akumulatorja predstavlja indeks tabele, na katerem se nahaja vrednost periode, ki jo PWM enota izstavi na izhod. Proces se ciklično ponovi vsakič, ko se sproži prekinitve, ki je določena s časovnikom, torej s frekvenco 31372,55 Hz. V primeru *Phase-Correct PWM* pride do prekinitve, ko časovnik doseže vrednost 0.

S krožnim potenciometrom, ki je povezan na Arduinov analogni vhod A1, se izbere osnovni signal.

```
waveform = analogRead(1)>>8;
```

`analogRead` nam vrne 10 bitno vrednost. Če nad to vrednostjo ( $2^{10}$ ) izvedemo pomik v desno za 8 mest, dobimo 2 najbolj pomembna bita s katerima določimo izbiro med štirimi ( $2^2$ ) osnovnimi signali. Implementirani so sinusni, pravokotni, žagasti signal in šum. Perioda vsakega signala je vzorčena z 256 vzorci. 8 bitne vrednosti period omenjenih signalov so shranjene v tabeli. Skupna velikost tabele je torej 1024 bajtov. Običajno se tabele shranijo v SRAM, kjer se ustvarjajo in upravljajo spremenljivke. Velikost SRAM-a v Arduino Uno je 2048 bajtov. Tabela bi torej zasedla polovico. Z ukazom *PROGMEM*, ki je del knjižnice *pgmspace.h*, shranimo tabelo v Flash spomin. Tako se izognemo prostorski stiski v SRAM-u.

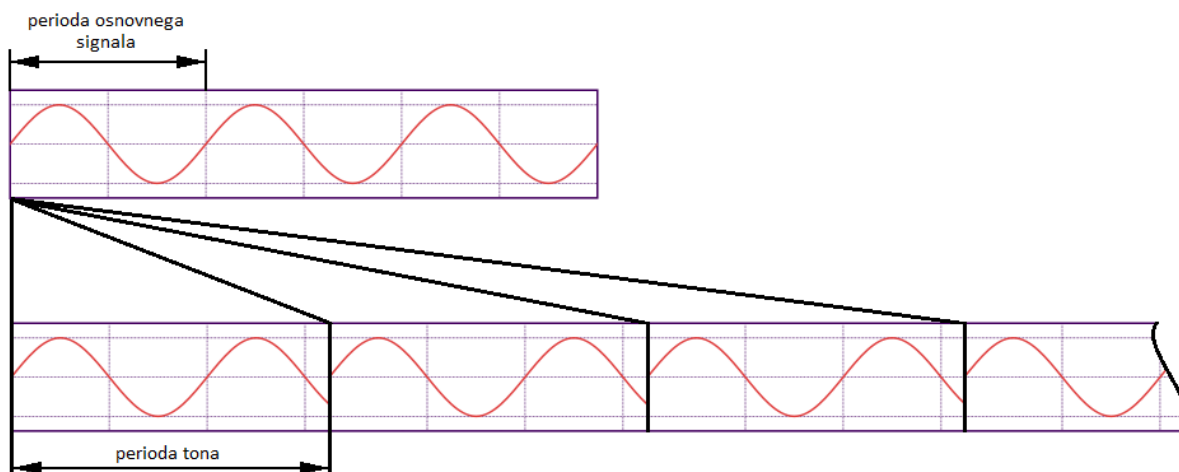


**Slika 9: Diagram poteka prekinitvenega servisnega programa**

Na Arduinov analogni vhod A0 je priključen potenciometer, s katerim določimo frekvenco osnovnega signala. Devet najbolj pomembnih bitov vrednosti potenciometra določi velikost inkrementa.

Na diagramu poteka prekinitvenega servisnega programa (Slika 9) vidimo kaj se zgodi ob vsaki prekinitvi.

Granularna sinteza je metoda za sintetiziranje zvoka [6]. Temelji na krajših koščkih zvoka dolžine približno od 1 do 50 ms imenovani »zrna« (Slika 10). Različna zrna je možno zložiti v plasteh, predvajati z različnimi hitrostmi, fazami, glasnostjo in frekvencami. Pri nižjih hitrostih predvajanja je rezultat zvočna »pokrajina«. Pri višjih hitrostih pa je slišati kot ton neobičajne barve.



Slika 10: Granularna sinteza – del osnovnega signala (zrno) predstavlja periodo tona

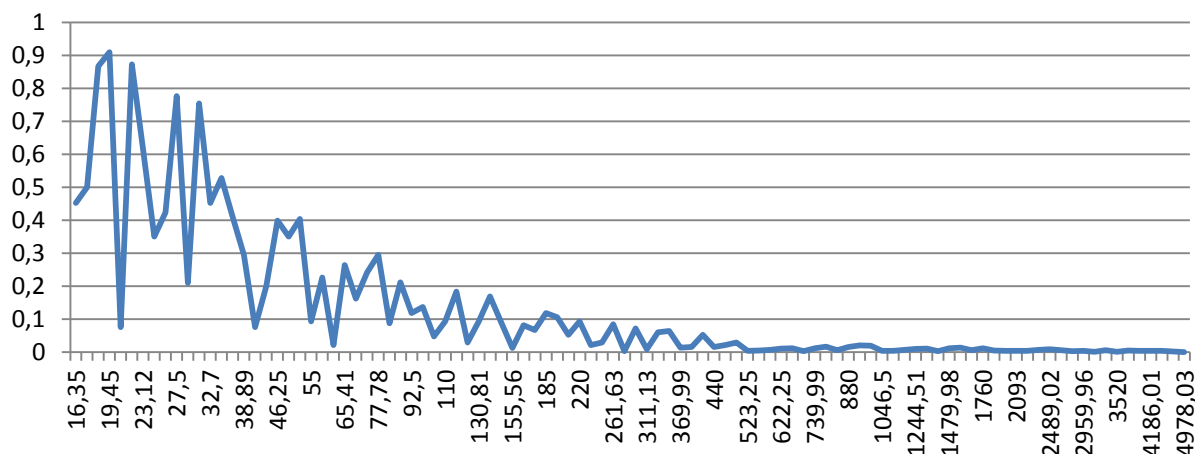
#### 2.2.4. Kromatična tabela inkrementov

Osnovni signal se z določeno frekvenco prekine in ponovno predvaja. Ta frekvenca je odvisna od tona, ki ga igramo na klaviaturi povezani preko MIDI kabla. Natančnost oziroma resolucija s katero določimo frekvenco ponavljanja osnovnega signala, je odvisna od akumulatorja in vzorčevalne frekvence  $F_s$ . Uporabili smo 16 bitni akumulator medtem, ko je  $F_s$  enaka 31372,55 Hz (izračunano v poglavju 2.2.1). Resolucija je torej:

$$\text{Resolucija} = \frac{\text{velikost akumulatorja}}{F_s}$$

$$\text{Resolucija} = \frac{2^{16}}{31372,55} = 2,08896$$

Inkrement za določeno frekvenco dobimo tako, da z njo množimo resolucijo. Zmnožek je potrebno zaokrožiti, ker je inkrement lahko le celo število. Zaokroževanje povzroči odstopanja dejanske frekvence v primerjavi z želeno. Slika 11 prikazuje ta odstopanja v procentih. Človek v povprečju loči nekaj procentna odstopanja pri frekvencah na območju do 100 Hz, pri 200 Hz loči približno 1% odstopanja (enakovredno šestini poltona), pri višjih frekvencah pa nekoliko manj [7]. To velja pri primerjavi dveh tonov, ki si sledita na zelo kratkem časovnem intervalu. V primeru dveh istočasno zaigranih tonov pa človek lahko loči zelo majhne razlike. Odstopanja našega sintetizatorja so pri 200 Hz desetina odstopanja, ki jih človek lahko zazna. Nad 500 Hz pa je napaka skoraj nična.

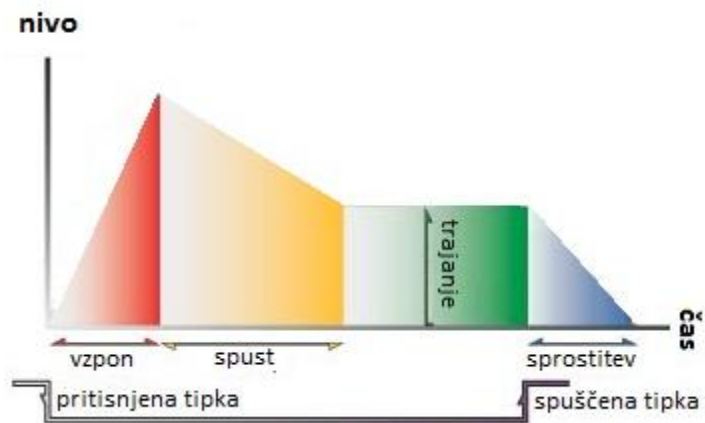


Slika 11: Odstopanja frekvenc v procentih

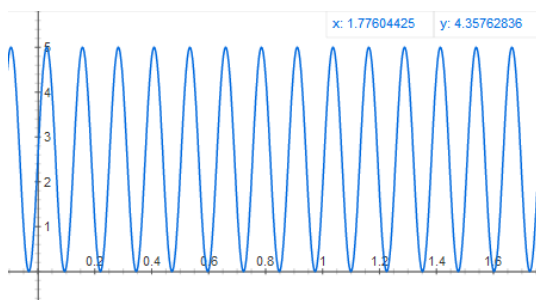
V priloženi tabeli (Tabela notnih frekvenc) lahko preberemo frekvence tonov in vrednost, ki jo MIDI klaviatura pošlje Arduino, ko pritisnemo tipko na njej. S pomočjo te tabele lahko izračunamo inkremente za vse tone in jih shranimo v kromatično tabelo inkrementov.

### 2.2.5. Prekinitveni servisni program

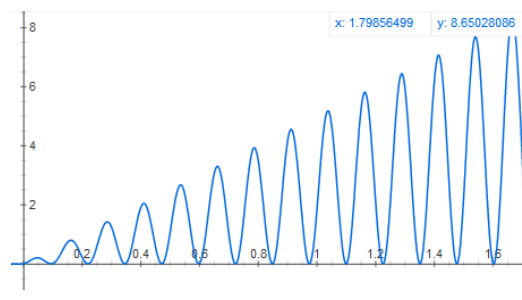
Prekinitveni servisni program (PSP) lahko razdelimo na tri ključne dele. V prvem delu se akumulatorjem prišteje vrednost ustreznega inkrementa. Osnovni signal in ton imata vsak svoj akumulator in inkrement. Akumulator osnovnega signala se postavi na 0, ko pride do preliva akumulatorja tona. V drugem delu se izračuna trenutno vrednost ovojnice signala (Slika 12). Ta je sestavljena iz vzpona (attack), spusta (decay), trajanja (sustain) in sprostitve (release). V tretjem delu se izračuna trenutno izhodno vrednost signala. Vrednost iz tabele osnovnega signala (Slika 13) množimo z vrednostjo ovojnice (Slika 14). Izhodni vrednosti je potrebno prišteti bias (Slika 15). Izhodni signal zvočnih naprav, kot je npr. prenosni mp3 player, je na linijskem nivoju. To je signal izmenične napetosti med -2,192V in +2,192V [8]. PWM izhod na Arduino ploščici je vrednost med 0V in 5V. Programsko to pomeni, da je na izhodu potrebna konstantna vrednost (bias), ki bo srednja vrednost za izmenični signal. Bias torej poveča vrednost izhodnega signala. To nam omogoča odštevanje polovične vrednosti ovojnice (Slika 16). Tako dosežemo simetričen signal, kar pomeni boljši izkoristek zvočnikov.



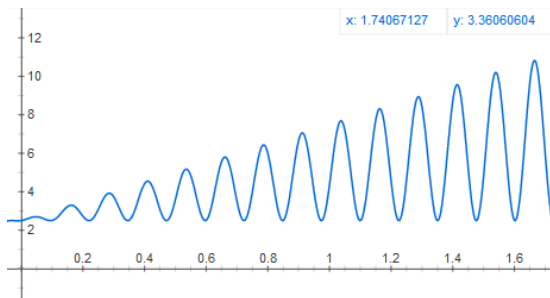
Slika 12: Ovojnica



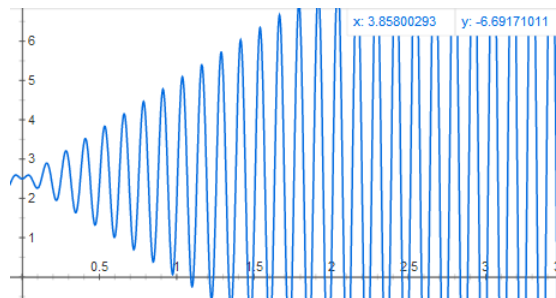
Slika 13: Sinusni osnovni signal



Slika 14: Sinusni osnovni signal zmnožen z ovojnico



Slika 15: Sinusni osnovni signal \* ovojnica + bias

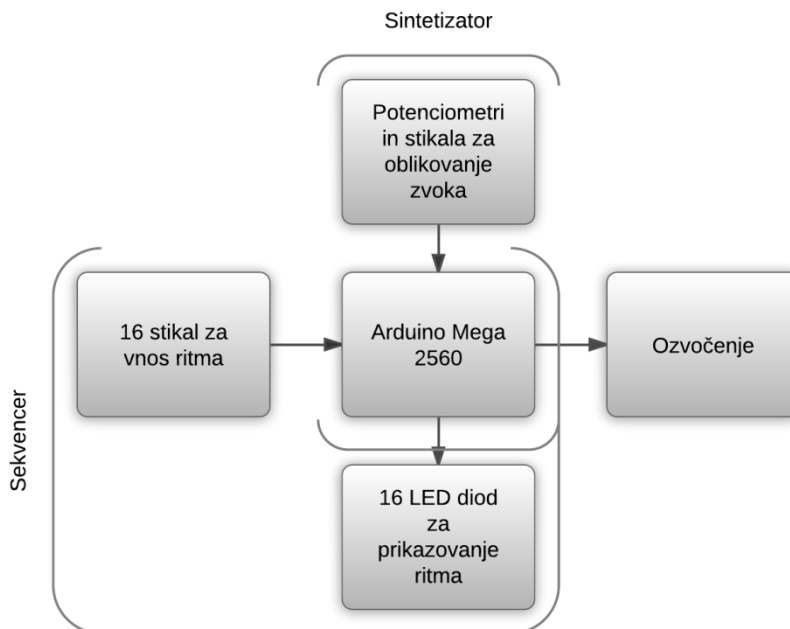


Slika 16: Sinus \* ovojnica + bias – ovojnica/2

## 3. Sekvencer in sintetizator bobnov

### 3.1. Vezje

Sekvencer in sintetiziranje bobnov zahtevata svojo ploščico Arduino zaradi zahtevnosti programa in števila potrebnih vhodov in izhodov. Sekvencer deluje tako, da s 16 stikali vnašamo ritem. 16 led diod kaže ritem in njegovo hitrost. Zvoke bobnov lahko spreminjamo s krožnimi potenciometri, glasnost izhodnega signala pa določimo z drsnim potenciometrom.

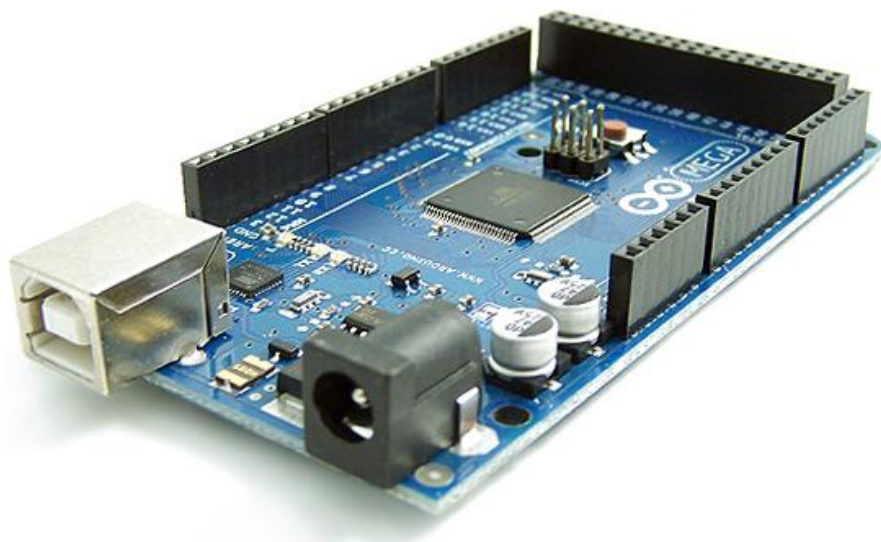


Slika 17: Blok shema vezja sekvencerja in sintetizatorja bobnov

#### 3.1.1. Arduino Mega 2560

Arduino Mega 2560 (Slika 18) je najzmogljivejša med ploščicami Arduino. Osnovana je na mikrokrmilniku ATmega2560. Razlog zaradi katerega smo jo izbrali, je veliko število digitalnih vhodov oz. izhodov. Ima jih 54. 15 digitalnih izhodov lahko uporabljamo kot PWM izhode. Analognih vhodov je 16 in jih lahko uporabimo tudi kot dodatne digitalne vhode oziroma izhode. Delovnega spomina SRAM je 8 KB. Mikrokrmilnik deluje s frekvenco 16MHz. Napaja in programira se na isti način kot Arduino Uno.

Na Arduino Mega 2560 je priključenih 16 preklopnih stikal, 3 osvetljena stikala, 16 led diod, 4 krožna potenciometra, 6.35 mm mono avdio priključek in drsni potenciometer za določanje glasnosti izhodnega signala.



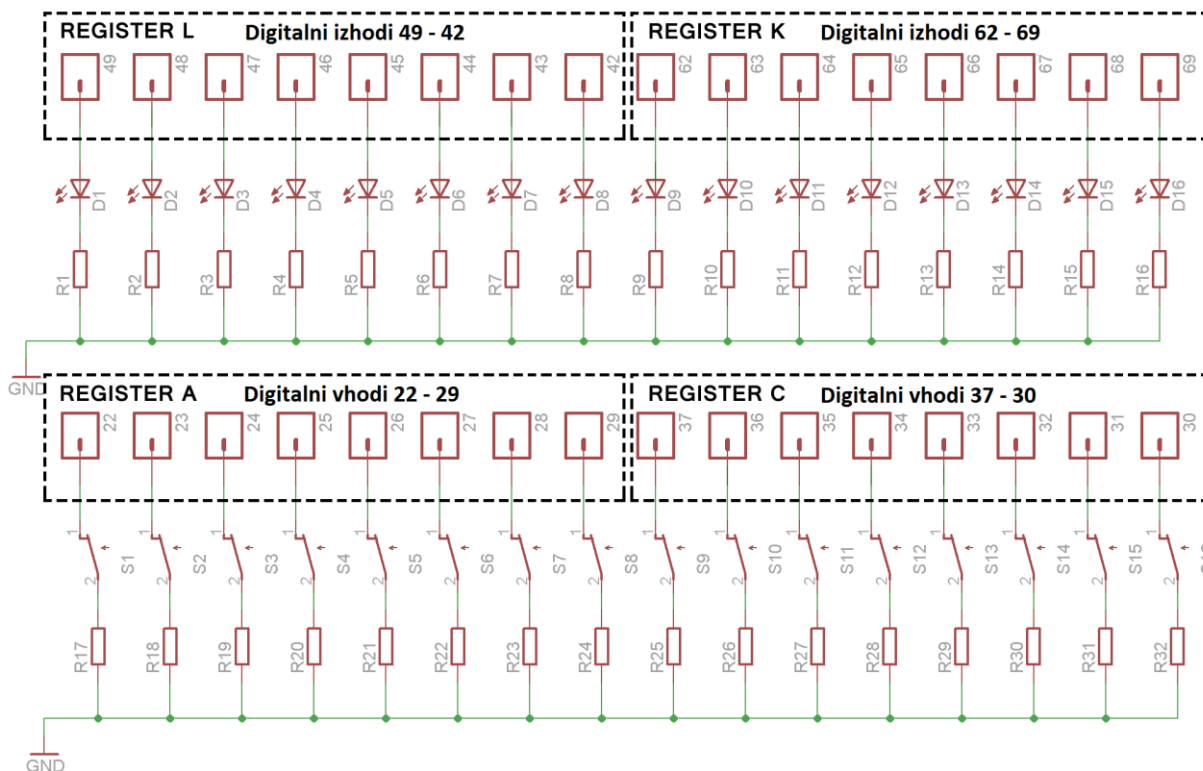
**Slika 18: Arduino MEGA 2560 razvojna ploščica**

### **3.1.2. Ostali elementi**

Štiri krožni linearni potenciometri z upornostjo  $4.7\text{ k}\Omega$  so povezani na maso (začetek potenciometra) in na +5V (konec potenciometra). Drsniki posameznih potenci metrov so priklopljeni na analogne vhode A0 do A3 na Arduino ploščici.

Drsni linearni potenciometer z upornostjo  $10\text{ k}\Omega$  je povezan na Arduinov digitalni izhod 3 (konec potenciometra) in na maso (začetek potenciometra). Nožica drsnika pa je povezana na 6.35 mm mono audio priključek. Potenciometer določa znižanje nivoja glasnosti izhodnega signala.

Tri osvetljena stikala so povezana na digitalne vhode 10 do 12. Stikalo poveže digitalne vhode z maso, ko je v pritisnjenem stanju. LED dioda v stikalu pa se takrat vklopi. LED diode in preklopna stikala, ki služijo vnašanju ritma so priklopljene na Arduinove digitalne vhode tako, kot je prikazano na shemi (Slika 19). Vsi upori imajo upornost  $1,1\text{ k}\Omega$ . Črtkani pravokotniki združujejo digitalne vhode oziroma izhode, ki jih je možno hkrati upravljati z manipulacijo posameznih registrov.



Slika 19: Shema LED diod in stikal za vnašanje ritma

### 3.2. Program

Program je napisan z istim razvojnim orodjem, ki smo ga uporabili za sintetizator zvoka. Tudi struktura programa je zelo podobna. Najprej vnesemo knjižnice za upravljanje vhodov in izhodov, prekinitve in programski spomin (Flash memory). Nato se deklarira globalne spremenljivke, konstante in tabele.

Po vklopu ploščice Arduino se zažene funkcija *setup()*. V tej funkciji določimo digitalne vhode 10-12, 22-29 in 30-38. Arduino vsebuje notranje dvižne upore, ki jih lahko vklopimo tako, da na želene digitalne vhode zapišemo visoko stanje. To storimo najhitreje z neposredno manipulacijo registrov vhodno/izhodnih vrat. V spodnji tabeli (Tabela 2) so zapisana vrata, ki smo jih uporabili in ustrezni digitalni vhodi/izhodi. Vsak bit registrov ustreza določenemu vhodu/izhodu. Zadnji so zapisani po vrstnem redu, od najmanj pomembnega bita do najpomembnejšega bita registra.

| Vrata | Digitalni vhodi/izhodi |
|-------|------------------------|
| A     | 22 – 29                |
| B     | 53 – 50, 10 – 13       |
| C     | 37 – 30                |
| K     | 62 – 69                |
| L     | 49 – 42                |

**Tabela 2: Vrata in ustrezni digitalni vhodi/izhodi**

Z registrom DDRx (Data Direction Register), kjer je x črka vrat, določimo smer. Logična ničla pomeni vhod, enka pa izhod. V registru PORTx, kjer je x črka vrat, zapišemo logično enko na bite katerih ustrezni vhod želimo priklopljen na dvizni upor.

Na koncu v funkciji *setup()* postavimo potrebne zastavice v registrih. Iz tabele izhodov časovnikov za Arduino Mega 2560 (Tabela 3) lahko razberemo, da moramo za uporabo PWM na nožici 3 uporabiti časovnik 3 (TCNT3). Postavljen bit WGM30 v TCCR3A določi način delovanja časovnika kot PWM Phase-Correct. Bit COM3C1 določi, da se izhod OC3B postavi v nizko stanje, ko se naraščajoč časovnik ujema s primerjalnim registrom in gre v visoko stanje, ko se ujema pri odštevanju časovnika.

| Izhod časovnika | Arduinov pin | Izhod časovnika | Arduinov pin |
|-----------------|--------------|-----------------|--------------|
| OC0A            | 13           | OC3C            | 3            |
| OC0B            | 4            | OC4A            | 6            |
| OC1A            | 11           | OC4B            | 7            |
| OC1B            | 12           | OC4C            | 8            |
| OC2A            | 10           | OC5A            | 44           |
| OC2B            | 9            | OC5B            | 45           |
| OC3A            | 5            | OC5C            | 46           |
| OC3B            | 2            |                 |              |

**Tabela 3: Razpored primerjalnih registrov po PWM izhodih na Arduino Mega 256**

V registru TCCR3B postavljen bit CS30 določi, da je urin delilnik 1.

Bit TOIE3 v registru TIMSK3 določa ali se bo sprožila prekinitev, ko bo prišlo do preliva v časovniku TCNT3. PSP se bo torej sprožil s frekvenco  $16\text{MHz}/510 = 31372,55\text{ Hz}$ .

**Register TCCR3A**

| Bit              | 7      | 6      | 5      | 4      | 3      | 2      | 1     | 0     |
|------------------|--------|--------|--------|--------|--------|--------|-------|-------|
| (0x90)           | COM3A1 | COM3A0 | COM3B1 | COM3B0 | COM3C1 | COM3C0 | WGM31 | WGM30 |
| Read/Write       | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W   | R/W   |
| Začetna vrednost | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     |

**Register TCCR3B**

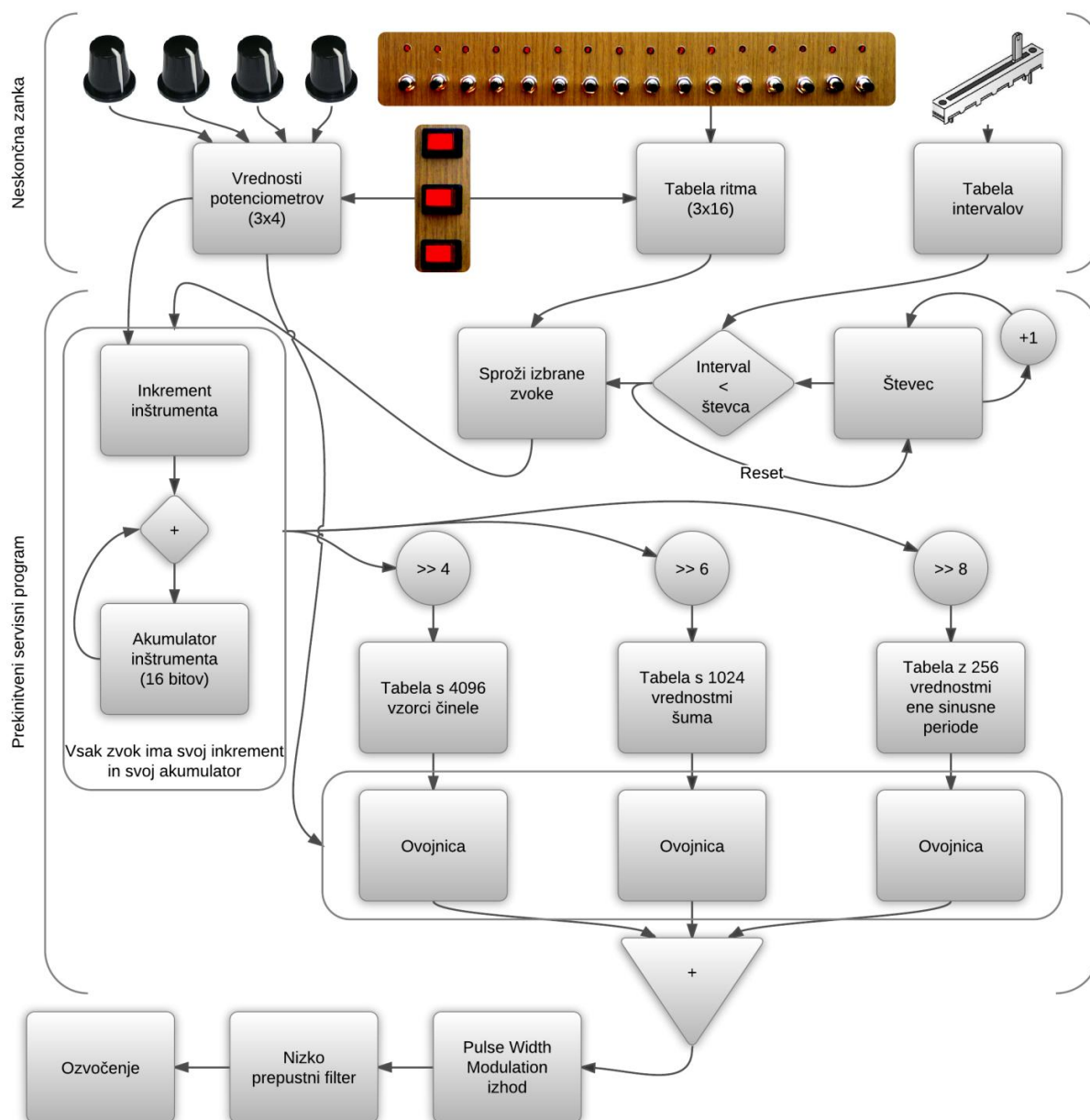
| Bit              | 7     | 6     | 5 | 4     | 3     | 2    | 1    | 0    |
|------------------|-------|-------|---|-------|-------|------|------|------|
| (0x91)           | ICNC3 | ICES3 | - | WGM33 | WGM32 | CS32 | CS31 | CS30 |
| Read/Write       | W/R   | W/R   | R | R/W   | R/W   | R/W  | R/W  | R/W  |
| Začetna vrednost | 0     | 0     | 0 | 0     | 0     | 0    | 0    | 0    |

**Register TIMSK3**

| Bit              | 7 | 6 | 5     | 4 | 3      | 2      | 1      | 0     |
|------------------|---|---|-------|---|--------|--------|--------|-------|
| (0x70)           | - | - | ICIE3 | - | OCIE3C | OCIE3B | OCIE3A | TOIE3 |
| Read/Write       | R | R | R/W   | R | R/W    | R/W    | R/W    | R/W   |
| Začetna vrednost | 0 | 0 | 0     | 0 | 0      | 0      | 0      | 0     |

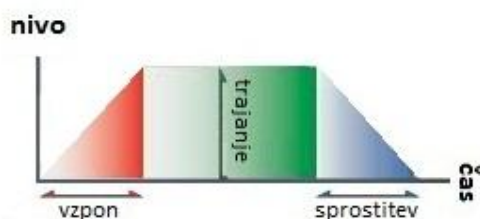
Po funkciji *setup()* se izvaja funkcija *loop()* v neskončni zanki. Tukaj se posodablja vrednosti potenciometrov in spremembe na digitalnih vhodih. Potenciometer na analognem vhodu A4 določa dolžino intervala s katerim si sledijo ritmični udarci. Vrednosti na digitalnih vhodih 10-12 določijo ali se spreminja parametre za bas boben, mali boben ali za činelo. Prav tako določijo od katerih treh zvokov naj LED diode prikazujejo ritem. Vrednosti potenciometrov na analognih vhodih A0-A3 oblikujejo zvok izbranega inštrumenta. Ritem, ki se vnese s trenutnimi stikali, se zabeleži v tabelo.

Diagram poteka funkcije *loop()* in PSP ritem mašine (Slika 20) prikazuje poenostavljeno delovanje programa. V PSP se poskrbi za zvok in za to da so LED diode prižgane. LED diode se zaporedoma vžigajo tako, da prikažejo hitrost, s katero se ritem predvaja in trenutno dobo v taktu (takt je razdeljen na 16 enakih intervalov). Diode, ki prikazujejo izbrani ritem, pa utripajo z nižjo frekvenco in svetijo medlo.



Slika 20: Diagram poteka funkcije `loop()` in PSP ritem mašine

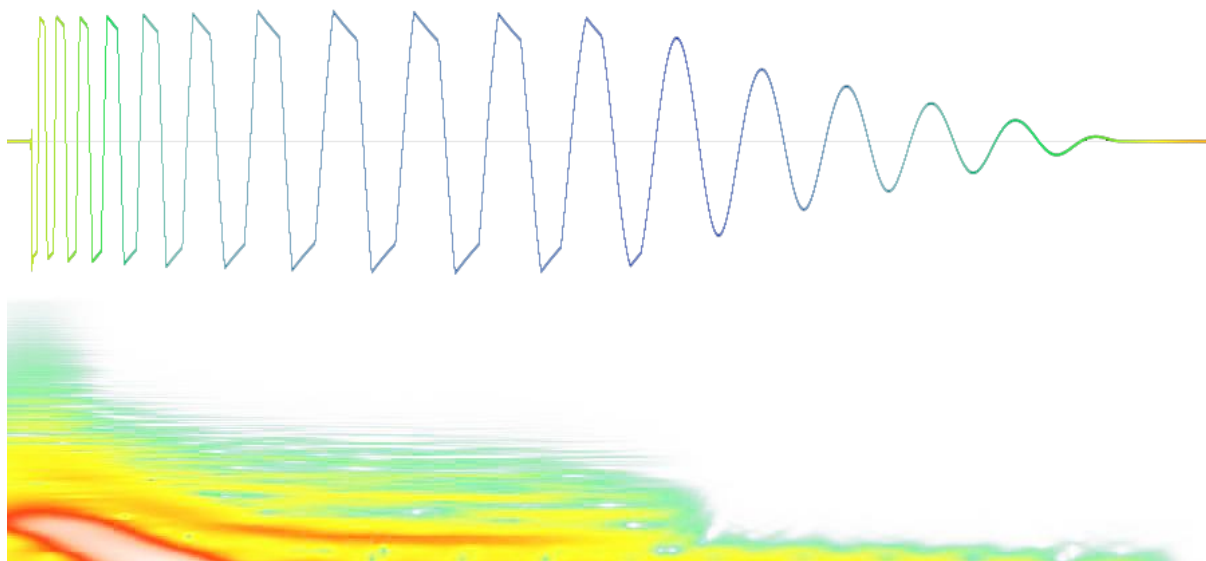
Zvok se sproži, ko je potekel interval do naslednje dobe v taktu. Od tistega trenutka pa do konca zvena zvoka se v vsakem ciklu PSP izračuna trenutno vrednost ovojnice. To velja za vsak zvok, ki je bil označen v tisti dobi v taktu. Ovojnica ritmičnih inštrumentov je bolj enostavna od ovojnice melodičnih. Ta zajema le vzpon, trajanje in sprostitev (Slika 21).



Slika 21: Ovojnica ritmičnega udarca (vzpon, trajanje, sprostitev)

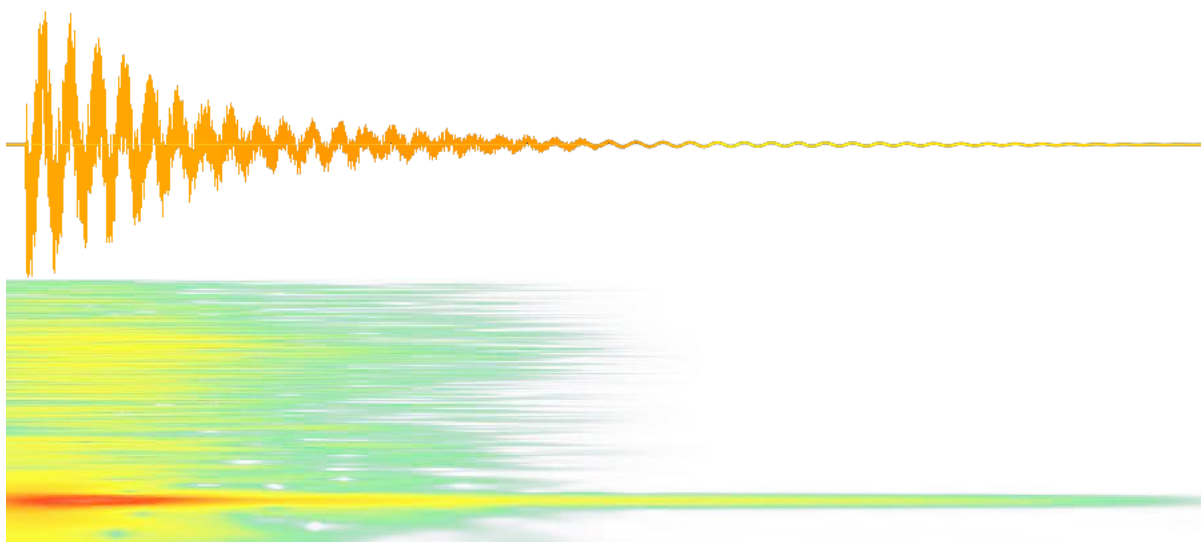
Ritmični zvoki lahko spreminjajo frekvenco tona v zvenu posameznega udarca. To je razvidno iz spektrograma (Slika 22). Za zgled smo vzeli zvoke Rolandove analogne ritem mašine TR909. Pri izračunu ovojnice smo zvok razdelili na tri dele (vzpon, trajanje in sprostitvev). Te bomo izkoristili za implementiranje treh faz frekvenčnega razvoja zvoka. Za sintetiziranje zvoka smo uporabili DDS tako kot pri sintetizatorju zvoka. V vsakem ciklu PSP se torej izračuna inkremente akumulatorjev.

Zvok bas bobna pri TR909 se začne z višjo frekvenco, ki predstavlja udarec po koži bobna. Za tem ton zvena pada dokler se ne ustali. Naša implementacija bas bobna je osnovana na sinusnem signalu. Frekvenca v fazi vzpona (traja 3 vzorce) je štirikratna končne. V fazi trajanja (traja 1023 vzorcev) frekvenca pade iz vrednosti določene s parametrom *toneDecay* na željeno frekvenco določeno s parametrom *grain\_freq\_ctrl*. Hitrost s katero frekvenca pada je konstantna. To pomeni, da bo padec iz višjih frekvenc trajal dlje. Daljši padec frekvence naredi zvok bolj »vesoljski«. Frekvenčni razpon našega bas bobna je omejen na tiste frekvence, ki so zanj smiselne. Parameter *grain\_freq\_ctrl* ima lahko vrednost med 450 in 962. Ta vrednost predstavlja indeks v tabeli *antilogTable[]*, kjer se nahaja vrednost inkrementa akumulatorja. V tabeli so shranjene 16 bitne vrednosti antilogaritmične funkcije. Nižja vrednost *grain\_freq\_ctrl* pomeni torej večji inkrement akumulatorja. Razlika med *grain\_freq\_ctrl* in *toneDecay* je lahko največ 962. Spremenljivka *toneDecay* se inkrementira za 1 v vsakem ciklu PSP. To pomeni, da se bo frekvenca ustalila najkasneje po 962 ciklih PSP v fazi trajanja. V fazi upada ni sprememb v frekvenci tona.



Slika 22: Valovna oblika (zgoraj) in spektrogram (spodaj) bas bobna Rolandove ritem mašine TR909

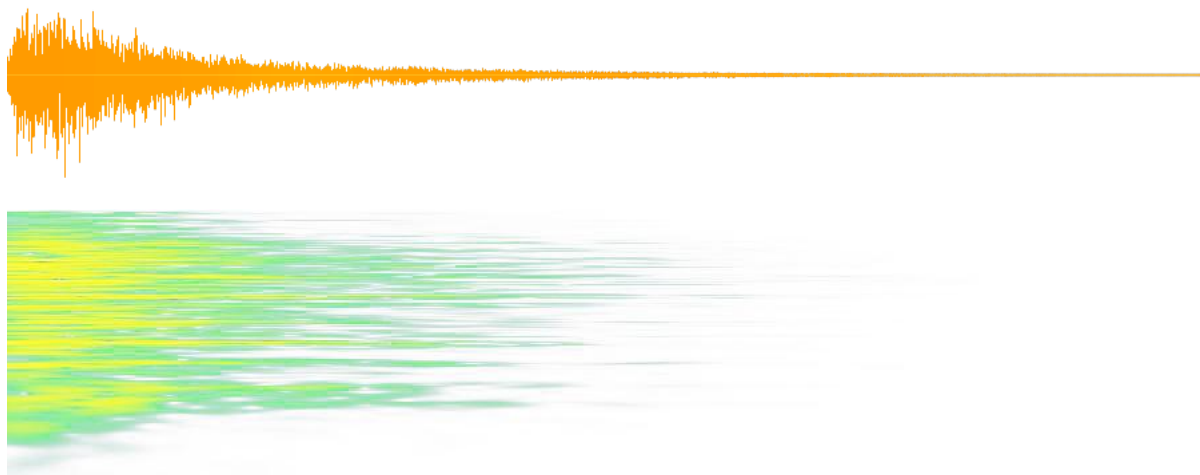
Spektrogram malega bobna Rolandove TR909 (Slika 23) kaže, da je zvok sestavljen iz dveh zvočnih komponent. Prva polovica udarca zajema večji del frekvenčnega spektra. To predstavlja mrežico na akustičnem malem bobnu. Po udarcu s palčko na boben, mrežica vibrira po koži bobna. To mu daje značilen zvok. Zvok mrežice lahko sintetiziramo s šumom, ki vsebuje vse frekvence v zvočnem frekvenčnem spektru. Na spektrogramu opazimo, da je del frekvenčnega spektra na začetku glasnejši (rdeče barve) in se s časom utiša (postane rumene barve, nato pa zelenomodre). To predstavlja drugo zvočno komponento oziroma ton malega bobna. Tega lahko sintetiziramo s sinusnim signalom. Naša implementacija sintetiziranja tona malega bobna je zelo podobna implementaciji bas bobna. Mali boben zveni z višjim tonom kot bas boben. Frekvenčni razpon na katerem lahko izbiramo frekvenco sinusa oziroma tona malega bobna, je primerno omejen. Parameter *grain\_freq\_ctrlSN* ima podobno vlogo kot *grain\_freq\_ctrl* pri bas bobnu. S potenciometrom določimo njegovo vrednost med 370 in 497. Kljub temu, da mali boben Rolandove TR909 nima frekvenčnega razvoja tona, smo se odločili za njegovo implementacijo. Tako lahko dosežemo »vesoljski« zvok tudi pri malemu bobnu. Parametri, ki jih lahko določimo, so dolžina upada šuma, dolžina upada tona, začetna frekvenca tona in končna frekvenca tona.



**Slika 23: Valovna oblika (zgoraj) in spektrogram (spodaj) malega bobna Rolandove ritem mašine TR909**

Štop činela Rolandove ritem mašine TR909 (Slika 24) ni sintetizirana z analognimi oscilatorji ampak uporabi vzorčeno činelo [9]. Naša implementacija je uporabila isto rešitev. V enodimenzionalno tabelo velikosti 4096 smo shranili 8 bitne vrednosti činele. Posnetek ne potrebuje dodatne ovojnice. Odločili smo se, da bomo lahko določali dolžino zvoka. V

primeru da činela ne odzveni do konca, jo je potrebno ustrezno zaključit. Implementirali smo fazi trajanja in upada. Trajanje faze upada je določeno na 1022 vzorcev oziroma ciklov PSP. Faza trajanja pa je spremenljiva in lahko traja od 0 do 4095 vzorcev.



**Slika 24: Valovna oblika (zgoraj) in spektrogram (spodaj) štop činele Rolandove ritem mašine TR909**

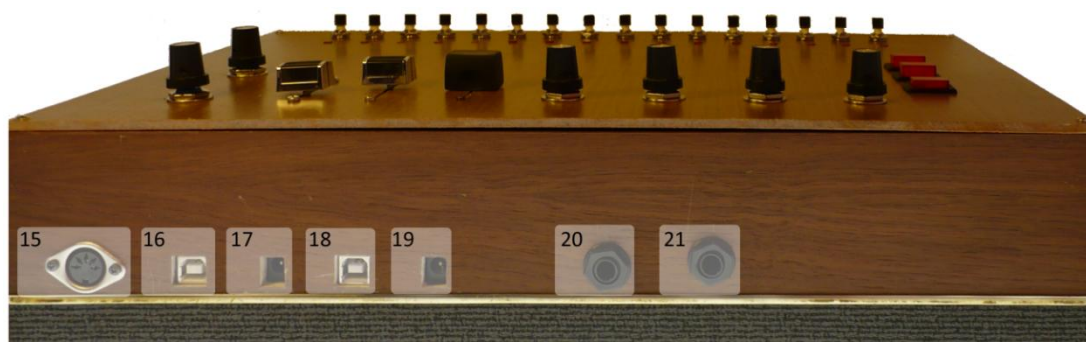
Izhodno vrednost izračunamo podobno kot pri sintetizatorju zvoka. Izhod brez zvoka ima vrednost biasa. 8 bitno vrednost iz ustrezne tabele množimo z ovojnico (6 bitno), da dobimo trenutno vrednost signala. Zmnožku odštejemo polovično vrednost ovojnice zaradi simetrije. Rezultat prištejemo izhodni vrednosti. Ta postopek uporabimo za zvok bas bobna in malega bobna. Vrednosti vzorčene činele že vsebujejo bias in ovojnico. Zaradi tega je potrebno izhodni vrednosti odšteti vrednost biasa vsakič, ko ji prištejemo vrednost iz tabele vzorčene činele.

## 4. Upravljanje inštrumenta

Sintetizator z MIDI vhodom in sekvencer sta fizično združena v istem ohišju. Za napajanje potrebujemo bodisi 2 USB kabla ali 2 adapterja enosmerne napetosti od 7V do 12V in s tokom približno 400mA. 2 avdio vtiča velikosti 6.35 mm sta ločena izhoda za sekvencer in sintetizator. Njuni masi je pomembno ohraniti ločeni, da se izognemo brnenju v signalu. Slika 26 prikazuje zadnjo stran ohišja kjer najdemo USB vtič sintetizatorja (16), vtič za napajanje sintetizatorja (17), USB vtič sekvencerja (18), vtič za napajanje sekvencerja (19), 6.35 mm mono avdio priključek sekvencerja (20) in sintetizatorja (21).



Slika 25: Zgornji del inštrumenta



Slika 26: Zadnji del inštrumenta

## 4.1. Upravljanje sintetizatorja

Za upravljanje sintetizatorja zvoka potrebujemo MIDI klaviaturo in MIDI kabel. S kablom povežemo klaviaturo z našim sintetizatorjem preko DIN konektorja 15 (Slika 26). Potenciometra 13 in 14 (Slika 25) na ohišju določata izhodni signal. S potenciometrom 14 določimo tip osnovnega signala, s potenciometrom 13 pa njegovo frekvenco. Na klaviaturi igramo poljubno melodijo. Drsní potenciometer 12 določa izhodno glasnost signala.

## 4.2. Upravljanje sekvencerja

Za upravljanje sekvencerja ne potrebujemo dodatne opreme. Ob priklopu napajanja se sintetizator prižge. LED diode (9) se prižigajo zaporedoma z leve proti desni. To pomeni, da sekvencer deluje in teče. S stikali 6, 7 in 8 (Slika 25) določimo kateri zvok bomo oblikovali. S stikalom 6 izberemo bas boben, s stikalom 7 izberemo mali boben, s stikalom 8 pa činelo. Ko vklopimo izbrano stikalo, se prižge LED dioda v njem. S stikali 10 (Slika 25) vnesemo ritem. Izhodni zvok lahko spreminjamo s potenciometri. V spodnji tabeli (Tabela 4) vidimo s katerimi parametri lahko oblikujemo posamezni zvok.

|            | Potenciometer 1       | Potenciometer 2 | Potenciometer 3        | Potenciometer 4 |
|------------|-----------------------|-----------------|------------------------|-----------------|
| Bas boben  | Končna frekvenca tona | Dolžina tona    | Začetna frekvenca tona |                 |
| Mali boben | Končna frekvenca tona | Dolžina tona    | Začetna frekvenca tona | Dolžina šuma    |
| Činela     | Frekvenca tona        | Dolžina tona    |                        |                 |

**Tabela 4: Parametri za oblikovanje posameznih zvokov.**

Hitrost ritma določimo z drsnim potenciometrom 5 (Slika 25). Glasnost izhodnega signala pa določimo s krajšim drsnim potenciometrom 11.

## 5. Sklepne ugotovitve

Izdelava inštrumenta je bila zelo zanimiva izkušnja, predvsem zaradi oblikovanja zvokov, ki jih predvaja. Ta del procesa je bil zelo kreativen in hkrati poučen. Okvirno sem že poznal obstoječe rešitve, vendar je bilo potrebno raziskovati detajle. To velja tako za sekvencer kot za MIDI protokol.

Večji problem je predstavljalo prilagajanje kvalitete zvoka in enostavnost uporabe zmogljivosti sistema. Pri obeh napravah je bilo potrebno paziti pri predvajanju več kot enega zvoka hkrati. Zaradi seštevanja vrednosti posameznih vzorcev se lahko na trenutke preseže največjo 8 bitno vrednost. V tem primeru pride do ostrega skoka z vrednosti 255 na 0. To se sliši kot pokanje v zvočniku, zato se je potrebno temu izogniti. Pri sintetizatorju, ki ima polifonijo 3, je bilo potrebno znižati končno vrednost posameznega zvoka. To smo storili tako, da smo zmanjšali njegovo bitno resolucijo za 2 bita. Tako smo rešili problem preliva v izhodnem signalu, vendar smo zmanjšali resolucijo na 6 bitov. Pri sintetiziranju bobnov je bilo potrebno prav tako žrtvovati 2 bita. Če je seštevek vseh vrednosti 8 bitni, se sliši na izhodu šum. Tega šuma ni, če na seštevkcu opravimo premik v desno za 1. To pomeni, da je končna resolucija dejansko 7 bitna.

Tehnične težave so včasih otežile delo. Kljub temu je bil opazen napredek na dnevni bazi, kar je bilo zelo spodbudno. Izdelava sintetizatorja ni zahtevala posebne ročne spretnosti. Pri izdelavi sekvencerja pa je bilo drugače. Zaradi velikega števila stikal in LED diod, je potrebno priklopiti veliko kablov. Pri sestavljanju ohišja moramo biti pozorni tudi na to, da se ti kabli ne odklopijo. Problem smo rešili z Arduinovim protoboard ščitom. To je ploščica, ki jo lahko priklopimo neposredno na digitalne in analogne vhode ter na vse ostale nožice na ploščici Arduino. Na ščit lahko spajkamo poljubne kable in čipe ter se tako izognemo neželenim izklopom kablov.

Programski del je bil precej bolj zahteven. Ploščica Arduino Mega 2560 je relativno nova na tržišču in ni zelo razširjena. Zanj je težko najti specifikacije. Specifikacije za Atmelov čip, ki je vanjo vgrajen, so lažje dosegljive. Problem nastane pri povezovanju digitalnih in analognih vhodov čipa z Arduinovimi. Na čipu si ne sledijo po istem vrstnem redu kot na ploščici Arduino, zato je težje ugotoviti, kateri registri predstavljajo določene vhode. Upravljanje digitalnih vhodov in izhodov preko registrov je bistveno hitrejše kot preko funkcije `digitalRead()` ali `digitalWrite()`. Če želimo zapisovat izhodno vrednost na PWM izhod

s frekvenco 31372,55 Hz, moramo ohraniti določeno hitrost programa. Pri phase correct PWM imamo na razpolago 510 urinih period za računanje vrednosti naslednjega vzorca. Zanimivo je, da prevajalnik razvojnega okolja Arduino ne prevede izvirne kode v najhitrejše možne strojne ukaze.

Prva stvar, ki bi jo bilo potrebno izboljšati v sistemu, je D/A pretvornik. PWM ni namenjen temu, vendar je cenejši. Tudi stikala za vnos ritma so se izkazala kot slaba izbira. Primernejša bi bila ploščata stikala, podobna tipkam računalniške tipkovnice, ker bi omogočala hitrejši vnos. Med delom so nastale ideje za dodatne funkcije sintetizatorja. Zelo zanimivo bi bilo, če bi lahko sami posneli periodo osnovnega signala. S senzorjem za razdaljo bi lahko beležili premik roke v določenem časovnem intervalu. Te vrednosti bi shranili v tabelo in bi jih lahko predvajali tako, kot periode ostalih signalov. Sekvencer bi lahko imel potenciometer za dodatno zmanjševanje resolucije zvokov. To bi dosegli tako, da bi izhodno vrednost premaknili desno za poljubno število bitov, za tem pa nazaj levo za isto število bitov. Biti bi se tako zbrisali in signal bi postal še bolj stopničast. Zelo uporabna bi bila možnost shranjevanja ritmov. Tako bi se lahko zaporedoma predvajalo več različnih ritmov. Trenutno ni možno igrati tri-četrtske takte. To bi lahko omogočili s potenciometrom, ki bi določal do katere šestnajstinke naj sekvencer teče. Ritem mašine imajo pogosto opcijo swinga. To pomeni, da se ritem na trenutke malenkost pohitri, na trenutke pa upočasni. Za tako implementacijo bi potrebovali dodaten programski oscilator. Zelo verjetno bi bilo to prezahtevno, ker je trenutni PSP na meji časovne zahtevnosti.

Rezultat je zanesljiv glasbeni inštrument, s katerim je možno samostojno nastopati, ker omogoča ustvarjanje in predvajanje tako ritma kot melodije. Znanje, ki sem ga pridobil pri izdelavi tega diplomskega dela, je dobra iztočnica za izdelavo boljših sistemov. Zanimivo je imeti nadzor nad vsakim vzorcem zvoka.

# Priloge

## 1. Tabela notnih frekvenc

| Nota                                                     | Frekvenca (Hz) | MIDI |
|----------------------------------------------------------|----------------|------|
| C <sub>0</sub>                                           | 16,35          | 12   |
| C <sup>#</sup> <sub>0</sub> /D <sup>b</sup> <sub>0</sub> | 17,32          | 13   |
| D <sub>0</sub>                                           | 18,35          | 14   |
| D <sup>#</sup> <sub>0</sub> /E <sup>b</sup> <sub>0</sub> | 19,45          | 15   |
| E <sub>0</sub>                                           | 20,6           | 16   |
| F <sub>0</sub>                                           | 21,83          | 17   |
| F <sup>#</sup> <sub>0</sub> /G <sup>b</sup> <sub>0</sub> | 23,12          | 18   |
| G <sub>0</sub>                                           | 24,5           | 19   |
| G <sup>#</sup> <sub>0</sub> /A <sup>b</sup> <sub>0</sub> | 25,96          | 20   |
| A <sub>0</sub>                                           | 27,5           | 21   |
| A <sup>#</sup> <sub>0</sub> /B <sup>b</sup> <sub>0</sub> | 29,14          | 22   |
| B <sub>0</sub>                                           | 30,87          | 23   |
| C <sub>1</sub>                                           | 32,7           | 24   |
| C <sup>#</sup> <sub>1</sub> /D <sup>b</sup> <sub>1</sub> | 34,65          | 25   |
| D <sub>1</sub>                                           | 36,71          | 26   |
| D <sup>#</sup> <sub>1</sub> /E <sup>b</sup> <sub>1</sub> | 38,89          | 27   |
| E <sub>1</sub>                                           | 41,2           | 28   |
| F <sub>1</sub>                                           | 43,65          | 29   |
| F <sup>#</sup> <sub>1</sub> /G <sup>b</sup> <sub>1</sub> | 46,25          | 30   |
| G <sub>1</sub>                                           | 49             | 31   |
| G <sup>#</sup> <sub>1</sub> /A <sup>b</sup> <sub>1</sub> | 51,91          | 32   |
| A <sub>1</sub>                                           | 55             | 33   |
| A <sup>#</sup> <sub>1</sub> /B <sup>b</sup> <sub>1</sub> | 58,27          | 34   |
| B <sub>1</sub>                                           | 61,74          | 35   |
| C <sub>2</sub>                                           | 65,41          | 36   |
| C <sup>#</sup> <sub>2</sub> /D <sup>b</sup> <sub>2</sub> | 69,3           | 37   |
| D <sub>2</sub>                                           | 73,42          | 38   |
| D <sup>#</sup> <sub>2</sub> /E <sup>b</sup> <sub>2</sub> | 77,78          | 39   |
| E <sub>2</sub>                                           | 82,41          | 40   |
| F <sub>2</sub>                                           | 87,31          | 41   |
| F <sup>#</sup> <sub>2</sub> /G <sup>b</sup> <sub>2</sub> | 92,5           | 42   |
| G <sub>2</sub>                                           | 98             | 43   |
| G <sup>#</sup> <sub>2</sub> /A <sup>b</sup> <sub>2</sub> | 103,83         | 44   |
| A <sub>2</sub>                                           | 110            | 45   |

| Nota                                                     | Frekvenca (Hz) | MIDI |
|----------------------------------------------------------|----------------|------|
| A <sup>#</sup> <sub>2</sub> /B <sup>b</sup> <sub>2</sub> | 116,54         | 46   |
| B <sub>2</sub>                                           | 123,47         | 47   |
| C <sub>3</sub>                                           | 130,81         | 48   |
| C <sup>#</sup> <sub>3</sub> /D <sup>b</sup> <sub>3</sub> | 138,59         | 49   |
| D <sub>3</sub>                                           | 146,83         | 50   |
| D <sup>#</sup> <sub>3</sub> /E <sup>b</sup> <sub>3</sub> | 155,56         | 51   |
| E <sub>3</sub>                                           | 164,81         | 52   |
| F <sub>3</sub>                                           | 174,61         | 53   |
| F <sup>#</sup> <sub>3</sub> /G <sup>b</sup> <sub>3</sub> | 185            | 54   |
| G <sub>3</sub>                                           | 196            | 55   |
| G <sup>#</sup> <sub>3</sub> /A <sup>b</sup> <sub>3</sub> | 207,65         | 56   |
| A <sub>3</sub>                                           | 220            | 57   |
| A <sup>#</sup> <sub>3</sub> /B <sup>b</sup> <sub>3</sub> | 233,08         | 58   |
| B <sub>3</sub>                                           | 246,94         | 59   |
| C <sub>4</sub>                                           | 261,63         | 60   |
| C <sup>#</sup> <sub>4</sub> /D <sup>b</sup> <sub>4</sub> | 277,18         | 61   |
| D <sub>4</sub>                                           | 293,66         | 62   |
| D <sup>#</sup> <sub>4</sub> /E <sup>b</sup> <sub>4</sub> | 311,13         | 63   |
| E <sub>4</sub>                                           | 329,63         | 64   |
| F <sub>4</sub>                                           | 349,23         | 65   |
| F <sup>#</sup> <sub>4</sub> /G <sup>b</sup> <sub>4</sub> | 369,99         | 66   |
| G <sub>4</sub>                                           | 392            | 67   |
| G <sup>#</sup> <sub>4</sub> /A <sup>b</sup> <sub>4</sub> | 415,3          | 68   |
| A <sub>4</sub>                                           | 440            | 69   |
| A <sup>#</sup> <sub>4</sub> /B <sup>b</sup> <sub>4</sub> | 466,16         | 70   |
| B <sub>4</sub>                                           | 493,88         | 71   |
| C <sub>5</sub>                                           | 523,25         | 72   |
| C <sup>#</sup> <sub>5</sub> /D <sup>b</sup> <sub>5</sub> | 554,37         | 73   |
| D <sub>5</sub>                                           | 587,33         | 74   |
| D <sup>#</sup> <sub>5</sub> /E <sup>b</sup> <sub>5</sub> | 622,25         | 75   |
| E <sub>5</sub>                                           | 659,26         | 76   |
| F <sub>5</sub>                                           | 698,46         | 77   |
| F <sup>#</sup> <sub>5</sub> /G <sup>b</sup> <sub>5</sub> | 739,99         | 78   |
| G <sub>5</sub>                                           | 783,99         | 79   |

| Nota                                                     | Frekvenca (Hz) | MIDI |
|----------------------------------------------------------|----------------|------|
| G <sup>#</sup> <sub>5</sub> /A <sup>b</sup> <sub>5</sub> | 830,61         | 80   |
| A <sub>5</sub>                                           | 880            | 81   |
| A <sup>#</sup> <sub>5</sub> /B <sup>b</sup> <sub>5</sub> | 932,33         | 82   |
| B <sub>5</sub>                                           | 987,77         | 83   |
| C <sub>6</sub>                                           | 1046,5         | 84   |
| C <sup>#</sup> <sub>6</sub> /D <sup>b</sup> <sub>6</sub> | 1108,73        | 85   |
| D <sub>6</sub>                                           | 1174,66        | 86   |
| D <sup>#</sup> <sub>6</sub> /E <sup>b</sup> <sub>6</sub> | 1244,51        | 87   |
| E <sub>6</sub>                                           | 1318,51        | 88   |
| F <sub>6</sub>                                           | 1396,91        | 89   |
| F <sup>#</sup> <sub>6</sub> /G <sup>b</sup> <sub>6</sub> | 1479,98        | 90   |
| G <sub>6</sub>                                           | 1567,98        | 91   |
| G <sup>#</sup> <sub>6</sub> /A <sup>b</sup> <sub>6</sub> | 1661,22        | 92   |
| A <sub>6</sub>                                           | 1760           | 93   |
| A <sup>#</sup> <sub>6</sub> /B <sup>b</sup> <sub>6</sub> | 1864,66        | 94   |
| B <sub>6</sub>                                           | 1975,53        | 95   |
| C <sub>7</sub>                                           | 2093           | 96   |
| C <sup>#</sup> <sub>7</sub> /D <sup>b</sup> <sub>7</sub> | 2217,46        | 97   |
| D <sub>7</sub>                                           | 2349,32        | 98   |
| D <sup>#</sup> <sub>7</sub> /E <sup>b</sup> <sub>7</sub> | 2489,02        | 99   |
| E <sub>7</sub>                                           | 2637,02        | 100  |
| F <sub>7</sub>                                           | 2793,83        | 101  |
| F <sup>#</sup> <sub>7</sub> /G <sup>b</sup> <sub>7</sub> | 2959,96        | 102  |
| G <sub>7</sub>                                           | 3135,96        | 103  |
| G <sup>#</sup> <sub>7</sub> /A <sup>b</sup> <sub>7</sub> | 3322,44        | 104  |
| A <sub>7</sub>                                           | 3520           | 105  |
| A <sup>#</sup> <sub>7</sub> /B <sup>b</sup> <sub>7</sub> | 3729,31        | 106  |
| B <sub>7</sub>                                           | 3951,07        | 107  |
| C <sub>8</sub>                                           | 4186,01        | 108  |
| C <sup>#</sup> <sub>8</sub> /D <sup>b</sup> <sub>8</sub> | 4434,92        | 109  |
| D <sub>8</sub>                                           | 4698,64        | 110  |
| D <sup>#</sup> <sub>8</sub> /E <sup>b</sup> <sub>8</sub> | 4978,03        | 111  |

## Seznam slik

|                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------|----|
| Slika 1: Blok shema sintetizatorja z MIDI vmesnikom .....                                                 | 5  |
| Slika 2: Arduino UNO razvojna ploščica .....                                                              | 6  |
| Slika 3: Optospojnik 6N138 .....                                                                          | 6  |
| Slika 4: Shema vezja za MIDI priključek .....                                                             | 7  |
| Slika 5: Krožna potenciometera (a), drsni potenciometer (b) in 6.35 mm mono avdio priključek (c) .....    | 7  |
| Slika 6: Duty Cycle pri PWM .....                                                                         | 8  |
| Slika 7: Nefiltriran in filtriran PWM signal .....                                                        | 9  |
| Slika 8: Phase-Correct PWM .....                                                                          | 10 |
| Slika 9: Diagram poteka prekinitvenega servisnega programa .....                                          | 13 |
| Slika 10: Granularna sinteza – del osnovnega signala (zrno) predstavlja periodo tona .....                | 14 |
| Slika 11: Odstopanja frekvenc v procentih .....                                                           | 15 |
| Slika 12: Ovojnica .....                                                                                  | 16 |
| Slika 13: Sinusni osnovni signal .....                                                                    | 16 |
| Slika 14: Sinusni osnovni signal zmnožen z ovojnico .....                                                 | 16 |
| Slika 15: Sinusni osnovni signal * ovojnica + bias .....                                                  | 16 |
| Slika 16: Sinus * ovojnica + bias – ovojnica/2 .....                                                      | 16 |
| Slika 17: Blok shema vezja sekvencerja in sintetizatorja bobnov .....                                     | 17 |
| Slika 18: Arduino MEGA 2560 razvojna ploščica .....                                                       | 18 |
| Slika 19: Shema LED diod in stikal za vnašanje ritma .....                                                | 19 |
| Slika 20: Diagram poteka funkcije <i>loop()</i> in PSP ritem mašine .....                                 | 22 |
| Slika 21: Ovojnica ritmičnega udarca (vzpon, trajanje, sprostitve) .....                                  | 22 |
| Slika 22: Valovna oblika (zgoraj) in spektrogram (spodaj) bas bobna Rolandove ritem mašine TR909 .....    | 23 |
| Slika 23: Valovna oblika (zgoraj) in spektrogram (spodaj) malega bobna Rolandove ritem mašine TR909 ..... | 24 |
| Slika 24: Valovna oblika (zgoraj) in spektrogram (spodaj) štop činele Rolandove ritem mašine TR909 .....  | 25 |
| Slika 25: Zgornji del inštrumenta .....                                                                   | 26 |
| Slika 26: Zadnji del inštrumenta .....                                                                    | 26 |

## Literatura

- [1] (2012) Arduino, Wikipedia. Dostopno na:  
<http://en.wikipedia.org/wiki/Arduino>
- [2] (2012) MIDI 1.0, Wikipedia. Dostopno na:  
[http://en.wikipedia.org/wiki/MIDI\\_1.0](http://en.wikipedia.org/wiki/MIDI_1.0)
- [3] (2012) Specifikacije za optospojnik 6N139. Dostopno na:  
<http://i2c2p.twibright.com/datasheet/6n139.pdf>
- [4] (2012) MIDI Library v3.2, Arduino Playground. Dostopno na:  
<http://arduino.cc/playground/Main/MIDILibrary>
- [5] (2004) Eva Murphy, Colm Slattery: »All about Direct Digital Synthesis«. Dostopno na:  
<http://www.analog.com/library/analogdialogue/archives/38-08/dds.html>
- [6] (2012) Granular synthesis, Wikipedia. Dostopno na:  
[http://en.wikipedia.org/wiki/Granular\\_synthesis](http://en.wikipedia.org/wiki/Granular_synthesis)
- [7] A. Frova, *Fisica nella musica*, Bologna: Zanichelli, 2010, pogl. 5.
- [8] (2012) Line level, Wikipedia. Dostopno na:  
[http://en.wikipedia.org/wiki/Line\\_level](http://en.wikipedia.org/wiki/Line_level)
- [9] (1984) TR-909 Specifications. Dostopno na:  
<http://www.synthage.com/downloads/TR-909.pdf>