

Ivan Dominić	0036465207
Viktor Čaplinski	0036488983
Jura Kralj	0036486520
Tomislav Nogić	0036491793
Matija Roglić	0036491057

SEMINARSKI RAD - SPVP



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradu informacija
Sveučilište u Zagrebu

D2M - Dance2Music



- Δ Umjetnicima, svima zainteresiranima
- Δ Osnove glazbe i rada u MAX/MSP i Wekinatoru
- Δ Upute za korištenje sustava
- Δ Programiranje Arduina, rad u Wekinatoru, MATLAB-u i Max/MSP-u

13. lipnja 2019



Sažetak

Razvijeni Dance2Music, skraćeno D2M, sustav namijenjen je i razvijan za sve osobe željne kreativnog izražaja putem glazbe i plesa. Sustav je namijenjen ponajprije plesačima, ali ne isključuje i sve druge zainteresirane osobe koje su voljne svojim pokretima i plesom proizvoditi glazbu. Takvi slični komercijalni sustavi već postoje, no ograničeni su na proizvodnju unaprijed određene glazbe. Naš je sustav zamišljen i napravljen tako da korisnik–pleslač može sam osmisliti svoju glazbenu koreografiju. Za korištenje D2M sustava potrebna su osnovna tehnička znanja te znanja rada s programima Wekinator i Max/MSP, koja se kroz korištenje lako savladaju i čija kompleksnost ovisi o kompleksnosti korisnikove izvedbe. Temeljni način rada sustava je da sa senzora akceleracije i žiroskopa podatke šalje na računalo, na kojem pomoću navedenih programa proizvodi glazbu. Takav dizajn sustava omogućuje korisniku praktički neograničenu slobodu kreacije. Nedostaci toga su nestabilnosti koje se javljaju zbog kompleksnosti i ograničene kvalitete komponenti.

Sadržaj

1. UVOD	3
2. OPIS SUSTAVA	4
3. KORIŠTENI ELEKTRONIČKI SKLOPOVI	5
3.1. iBeacon Station	5
3.2. WEMOS LOLIN32	6
3.3. Ostali korišteni uređaji	6
4. REALIZACIJE SUSTAVA	7
4.1. MATLAB	7
4.2. Wekinator	8
4.3. Max/MSP	8
5. ZAKLJUČAK	12
6. LITERATURA	13
7. POJMOVNIK	14

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Svakom umjetniku je poznato da je prilikom svog izričaja ograničen faktorima koji određuju njegovo stvaranje i kreativnost. Tako je, na primjer, slikar ograničen paletom boja i veličinom platna, a plesač glazbom na koju pleše i njenom temom. D2M sustav osmišljen je tako da korisniku-plesaču daje novu dimenziju slobode stvaranja i umjetničkog izražavanja.

Sustav je konkretno koncipiran tako da plesač sam osmisli svoju glazbenu koreografiju, no to ne ograničava početnike i „obične ljude“ da se okušaju u plesu, već naprotiv, potiče da se u tome i okušaju. Zamišljeni sustav posebno je bitan plesačima koji se osjećaju ograničeno klasičnim plesovima i popratnom glazbom jer korištenjem D2M oni sami svojim plesom stvaraju glazbu.

Značajnim padom cijena kompletnih ugradbenih sustava moguće je gotovim (*off-the-shelf*) komponentama napraviti naš zamišljeni D2M sustav. Samim time stvorena je mogućnost pobližeg spajanja tehnologije i umjetnosti i time pružanja jedinstvenog iskustva publici i izvođaču.

Rješenja slična ovome već postoje i mogu se komercijalno nabaviti. Korištenjem već gotovih sustava korisnik pomoću pokreta može stvarati određene zvukove ili glazbu, no ograničen je tvorničkim postavkama i u pravilu nije ih u mogućnosti mijenjati.

Ako korisnik odluči ne koristiti ovaj sustav ostaje mu mogućnost korištenja već gotovih rješenja koja su potencijalno skuplja i najvjerojatnije ne pružaju toliku slobodu kreacije kao D2M. Također, moguće je razvijati vlastiti sustav što bez prethodnog predznanja može biti dosta zahtjevno. Krajnja opcija je naravno ples izvoditi bez korištenja ikakvog sustava, na uobičajeni način, s postojećom glazbom.

2. Opis sustava

Sustav je zamišljen tako da se bazira na maksimalno dva senzora koji detektiraju promjenu akceleracije i žiroskopski senzor. Senzori bi se trebali moći nositi na rukama i ne ometati plesača. Zbog jednostavnosti izvedbe i nemogućnosti spajanja oba senzorska modula u isto vrijeme korišten je samo jedan. Senzorski modul podatke sa senzora šalje putem Bluetooth veze na mikrokontroler koji primljene podatke šalje serijskom vezom na računalo. Na računalo podatci se primaju pomoću MATLAB skripte i šalju na program Wekinator. Na njemu se pomoću zadanih vrijednosti podatci obrađuju i šalju na program Max/MSP u kojemu se vrši konačna pretvorba podataka u zvuk.

3. Korišteni elektronički sklopovi

3.1. iBeacon Station

iBeacon Station, prikazan na slici 1, je BLE multisenzorski modul koji sadrži:

- Nordic nRF51822 BLE i 2.4 GHz SoC,
- TDK MPU6050 troosni akcelerometar i žiroskop,
- Bosch BMP180 senzor temperature i tlaka,
- Dyna Image AP3216 senzor ambijentalnog osvjetljenja i pokreta.

Veličina modula je 30 mm x 30 mm te se može napajati CR2032 baterijom napona 3 V. Zbog malih dimenzija modul je praktički neprimjetan i ne ograničava pokrete plesača.



Slika 1. iBeacon Station

Pri izradi ovog projekta korišteni su nRF51822 i MPU6050.

nRF51822 je SoC vrlo niske potrošnje namijenjen korištenju u BLE i 2,4 GHz bežičnim aplikacijama. Sastoji se od ARM Cortex-M0 procesora, 256 kB *flash* memorije, 32 kB RAM-a i radija. Uz to sadrži i mnoštvo digitalnih i analognih perifernih sklopova te mogućnost povezivanja s ostalim sklopovima putem digitalnih sučelja kao što su SPI, I²C ili UART.

MPU 6050 u jednom čipu integrira MEMS akcelerometar i žiroskop te procesor. Sadrži šest analogno-digitalnih pretvornika, kojima se digitaliziraju izlazi akcelerometra i žiroskopa, nakon čega se I²C sabirnicom šalju na nRF51822, koji ih preko Bluetooth-a prosljeđuje prijatelju povezanom na računalo.

Modul tvornički dolazi konfiguriran kao *beacon*, s *firmwareom* koji preko Bluetooth-a šalje podatke s akcelerometra i žiroskopa frekvencijom od 6 Hz.

3.2. WEMOS LOLIN32

Kako bi omogućili primanje podatke na računalu i vršenje daljnje obradu nad njima, bio je potreban uređaj koji bi služio kao veza između BLE senzora i računala. Radi jednostavnosti programiranja, izabrana je WEMOS LOLIN32 razvojna pločicu za ESP32 SoC, prikazana na slici 2.



Slika 2. WEMOS LOLIN32

Pomoću Arduino razvojne okoline, na WEMOS LOLIN32 razvojnoj pločici, implementiran je BLE klijent koji se spaja na nRF51822 senzorski modul. Senzorski modul koristi modificirani NUS (*Nordic Uart Service*) pa je omogućavanje notifikacija pojednostavljeno.

Program skenira BLE uređaje te se spaja na prvi pronađeni uređaj s NUS servisom. Nakon uspješnog spajanja na željeni uređaj, u glavnoj petlji, omogućavaju se notifikacije. Nakon primljene notifikacije, poziva se funkcija koja šalje primljene podatke na serijski port. Primaju se notifikacije podataka s akcelerometra i žiroskopa a zatim se na serijski port šalje slovo A ili G i vrijednosti triju osi akcelerometra ili žiroskopa. Radi lakšeg primanja podataka, svaki podataka (uključujući slova A i G) šalje se u novom retku.

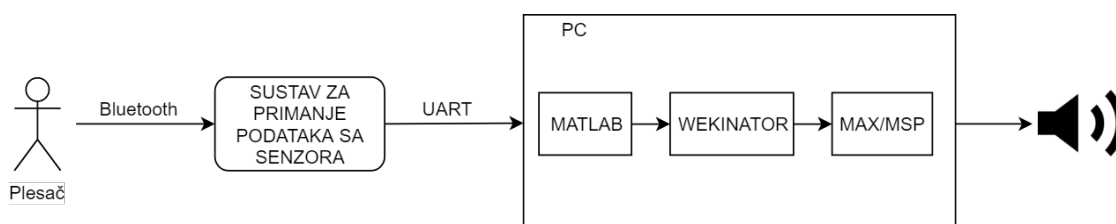
Program prima samo podatke s 2 senzora, ali moguće ga je jednostavno modificirati da prima i podatke s drugih senzora koji se nalaze senzorskom modulu.

3.3. Ostali korišteni uređaji

Uz prethodno opisani senzorski čvor i mikrokontroler, potrebno je imati računalo s instaliranim MATLAB, Wekinator i Max/MSP programskim paketima te na računalo spojiti zvučnik za reprodukciju proizvedenog zvuka.

4. Realizacije sustava

Blokovska shema realizacije kompletnog sustava prikazana je na slici 3.



Slika 3. Blokovska shema kompletnog sustava

Podaci sa senzora pristižu na računalo putem serijske veze. Koristeći MATLAB-ove funkcije za rad sa serijskim sučeljem, pristigli podaci se čitaju te na odgovarajući način pretvaraju u format prikladan za daljnju obradu pomoću Wekinatora (OSC) i stvaranje zvuka pomoću Max/MSP-a.

4.1. MATLAB

MATLAB je korišten zbog toga što je vrlo fleksibilan i jednostavan te sadrži široku paletu već implementiranih funkcija. U ovom projektu MATLAB je korišten za primanje podataka s mikrokontrolera, njihovu pretvorbu u format koji koristi Wekinator te slanje OSC poruka Wekinatoru putem UDB-a. Programska podrška u MATLAB-u napisana je u obliku MATLAB skripte.

Na početku se inicijalizira serijska veza između mikrokontrolera i računala te kreira UDP objekt preko kojega će se podaci primljeni serijskom vezom slati na daljnju obradu pomoću Wekinatora.

Zatim se čeka da se uspostavi veza između mikrokontrolera i BLE modula i pri tome se koristi ispis na komandu liniju kako bi se mogli uočiti eventualni problemi prilikom povezivanja. Nakon uspješne inicijalizacije i uspostavljanja veze počinje prijenos podataka s akcelerometra i žiroskopa. U beskonačnoj petlji se čitaju podaci koji pristižu u formatu: „A“, podaci akcelerometra, „G“, podaci žiroskopa. Podaci se ispisuju na komandnu liniju MATLAB-a, nakon čega se predaju kao argument funkciji `oscsend`, koja ih pretvara u OSC format te ih putem UDP-a šalje na Wekinator.

Kada se želi prekinuti prijenos, potrebno je zaustaviti izvođenje skripte i terminirati vezu s mikrokontrolerom te obrisati objekt pomoću kojeg se upravljalo serijskim i UDP prijenosom.

4.2. Wekinator

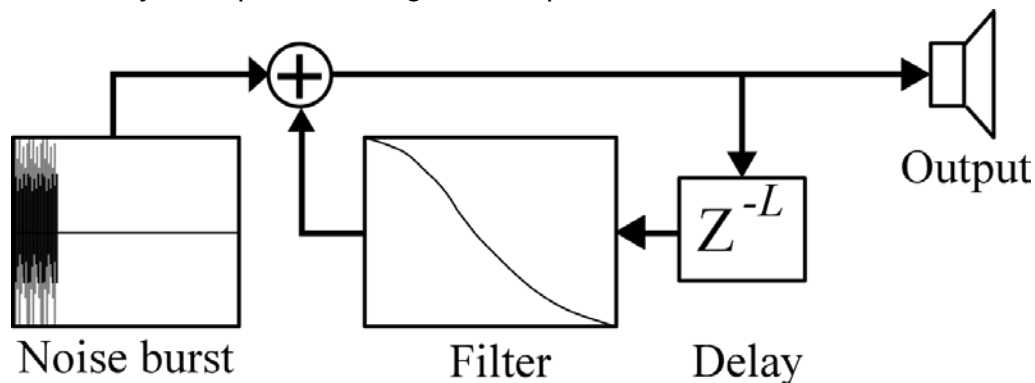
Wekinator je programska potpora otvorenog koda koja nam omogućava strojno učenje u stvarnom vremenu. Prilikom izrade novog modela potrebno je definirati broj ulaznih i izlaznih podataka. S obzirom na to da senzorski modul šalje podatke sa žiroskopa i akcelerometra u x, y i z ravnini, ukupni broj ulaznih parametara je 6. Broj izlaznih parametara je proizvoljan te je u konačnom modulu 5 izlaznih parametara koji kontroliraju frekvenciju gitare, tempo i 3 za svaki bubanj zasebno. Valja napomenuti da u konačnoj verziji nije implementiran treći bubanj ali je ostavljena mogućnost njegove jednostavne implementacije i izbora pokreta koji će aktivirati pokret bubnja.

Kako bi se uvježbao model na primjeru bubnjeva odabrani su pokreti za bubanj: pokret rukom sa senzorom u smjeru prema dolje kao da palicom udaramo bubanj te udarac u prsa otvorenom šakom. Bitno je imati jasno distinktivne pokrete kako bismo spriječili da udari krivi bubanj zbog sličnosti pokreta. Jednom kad je pokret bio jasno definiran bilo je potrebno model istrenirati da se proizvede zvuk udarca bubnja. Wekinator na izlaz šalje *float* varijablu u rasponu od 0 do 1 koju potom Max program prima te kada vrijednost te varijable pređe 0.7 proizvede zvuk udarca bubnja. Kako bi model naučio koji pokret je potrebno izvesti da se proizvede zvuk bubnja prvo su definirani svi neželjeni položaji te je za njih postavljen uvjet da izlazna vrijednost mora biti nula. Potom su snimljeni ulazni podatci sa senzora za vrijednost nula, za to vrijeme korisnik drži rukavicu u mirnom položaju i ostalim pozicijama koje nisu željene. Nakon što model nauči da u tim položajima mora držati nisku izlaznu vrijednost, postavljamo izlaznu vrijednost na 1 te snimamo samo željeni pokret 10 do 20 puta. Prilikom ovih snimanja bitno je da korisnik ne vraća ruku u prvotni položaj jer ćemo dobiti dvostruki udarac, jednom kad korisnik spušta ruku prema dolje a drugi put prema gore. Nakon nekoliko snimanja treniramo model i provjeravamo rezultat. U slučaju da primijetimo neželjeno ponašanje, korigiramo ga s novim snimkama. Proces ponavljamo dok ne dobijemo željeno ponašanje u više od 95% slučajeva. Postupnim snimanjem bi se mogla još više povećati točnost programa. Za frekvenciju gitare definiran je položaj ruke tako da podizanjem ruke stvaramo više tonove i obrnuto.

4.3. Max/MSP

Max/MSP je vizualni programski jezik stvoren za podršku glazbenicima i umjetnicima u jednostavnom stvaranju vlastitih instrumenata. Jednostavnost korištenja i pisanja programskih isječaka te podešavanje pojedinih elemenata čine ovaj programski jezik idealnim za

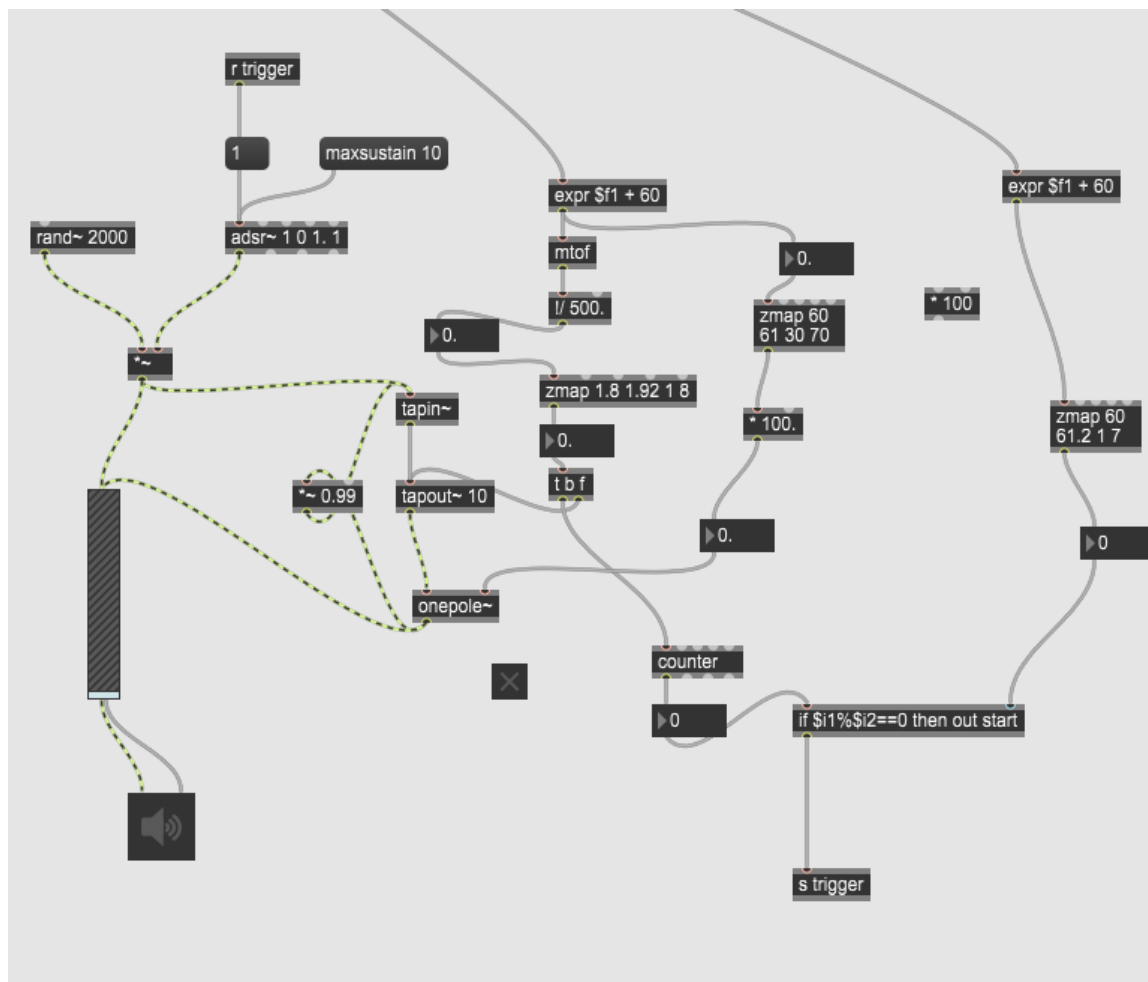
obradu izlaznih signala i stvaranje instrumenata. Kako bi se stvorio zvuk gitare iskorišten je Karplus-Strong model prikazan na slici 4.



Slika 4. Karplus-Strong model

Kao ulazni podatak, Karplus-Strong model prima signal stvoren kao umnožak ADSR signala i bijelog šuma. ADSR signal predstavlja oblik zvuka te se sastoji od četiri faze: *Attack*, *Decay*, *Sustain* i *Release*. Ovaj signal je jedan od najčešćih upotrebljavanih u standardnim sintetizatorima zvuka. Primljeni signal ulazi u sustav s povratnom vezom koja isti signal zakasni i filtrira. Kao izlaz dobivamo zvuk sličan trzaju na gitari. U povratnoj vezi možemo kontrolirati frekvenciju nekim vanjskim parametrom što i činimo pomoću Wekinator modela.

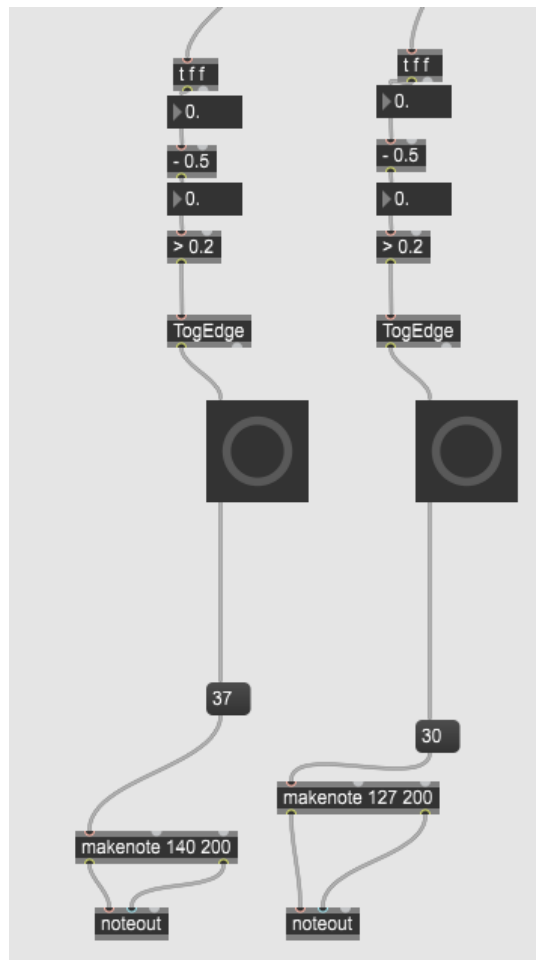
Implementacija Karplus-Strong modela u Max/MSP-u prikazana je na slici 5.



Slika 5. Implementacija Karplus-Strong modela u Max/MSP-u

Izvedba bubnjeva u Max/MSP-u je iznimno jednostavna. Postavljeni su uvodni uvjeti koji određuju kada će se poslati takozvani *bang* koji će simulirati udarac bubnja. Kako je već spomenuto ranije u dokumentaciji, da bi bubanj udario, potrebno je da ulazna varijabla prijeđe vrijednost 0.7. Jednom kad prijeđe tu vrijednost pušta se MIDI nota koja na izlaz pušta jedan udarac.

Implementacija bubnjeva prikazana je na slici 6.



Slika 6. Implementacija bubnjeva u Max/MSP-u

Naposljetku, komunikacija sa Max/MSP-om je izvedena preko standardnog porta za Wekinator: 12000. Na početku je bitno definirati objektom *unpack* koliko se varijabli prenosi te kojeg su tipa. Jednom kad su izlazne varijable definirane, moguće je poslije na njima vršiti obradu kako bi upravljale ostalim dijelovima Max programa.

5. ZAKLJUČAK

Razvijeni sustav Dance2Music, D2M, zamišljen je i razvijan s namjerom da omogući korisniku stvaranje jedinstvene glazbe pomoću njegovih pokreta ruku. Sustav se bazira na nRF51288 multisenzorskom modulu, WEMOS LOLIN32 razvojnom sustavu baziranom na ESP32 mikrokontroleru, te programima MATLAB, Wekinator i Max/MSP.

Pomoću razvijenog sustava uspješno su primljeni podaci akcelerometra i žiroskopa poslani preko Bluetooth-a, te su proslijeđeni na računalo putem serijske veze. Podaci su zatim na računalu primljeni u MATLAB-u, gdje su obrađeni i proslijeđeni na Wekinator, koji ih obrađuje u skladu s istreniranom karakteristikom te ih šalje na razvijeni program u Max/MSP-u.

Uspješno je implementiran sustav pomoću jednog senzorskog modula. Spajanje još jednog senzorskog modula bilo je planirano, ali se zbog poteškoća pri implementaciji od toga odustalo. Također se, nakon neuspjelih pokušaja da se senzorski moduli spajaju na BLE 400 Bridge, odustalo od korištenja tog Bridge-a i umjesto njega korišten je LOLIN32, koji se pokazao puno jednostavnijim za programiranje.

Od ovako izvedenog sustava imat će korist plesači koji žele izvesti jedinstvenu predstavu i koji su spremni sami proučiti rad Max/MSP programa jer je jedino tako moguće proizvesti glazbu u potpunosti po svojoj želji. Ostatak sustava je dovoljno uključiti i on automatski radi (*plug'n play*).

Daljnja želja je dodati sustavu još jedan senzor za drugu ruku, što je potrebno programski riješiti na kontroleru i na senzorskom modulu. Također bi bilo poželjno povećati brzinu slanja podataka sa senzora na više od 6 Hz kako bi se dobio uglađeniji prijelaz za vrijeme naglih pokreta.

6. Literatura

- [1] Arduino, URL <http://en.wikipedia.org/wiki/Arduino>, (12. 6. 2019.)
- [2] MAX 8 tutorial, URL <https://www.youtube.com/user/dude837> (12. 6. 2019.)
- [3] Marijnissen, Mark: oscsend MATLAB funkcija, URL <https://www.mathworks.com/matlabcentral/fileexchange/31400-send-open-sound-control-osc-messages>, (10. 5. 2019)
- [4] Nordic Semiconductor, "NRF 5 series", NRF51288 tehnička dokumentacija, URL https://infocenter.nordicsemi.com/pdf/nRF51822_PB_v2.5.pdf, (1. 4. 2019.)
- [5] Wekinator, tutoriali, URL <http://www.wekinator.org/>, (10. 4 .2019.)
- [6] WEMOS LOLIN32 razvojno sučelje, upute za rad, URL https://wiki.wemos.cc/products:lolin32:lolin32#technical_specs, (26. 4. 2018.)

7. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Bluetooth	Bežični standard za razmjenu podataka između mobilnih i fiksnih uređaja	https://en.wikipedia.org/wiki/Bluetooth
Bluetooth beacon	Mali Bluetooth odašiljač napajan baterijom	https://www.beaconstac.com/what-is-a-bluetooth-beacon
I ² C	Sučelje za sinkronu serijsku komunikaciju	https://en.wikipedia.org/wiki/I%C2%B2C
Microelectromechanical systems (MEMS)	Mikroskopski uređaji sa pokretnim dijelovima	https://en.wikipedia.org/wiki/Microelectromechanical_systems
Off-the-shelf	Komercijalno rješenje proizvoda koje je napravljeno da zadovolji potrebe kupovne organizacije, a ne po narudžbi	https://en.wikipedia.org/wiki/Commercial_off-the-shelf
Plug and play	Izraz za sustav fizički ili programski koji je potrebno samo upaliti kako bi normalno radio bez potrebe korisnikove konfiguracije	https://en.wikipedia.org/wiki/Plug_and_play
Random-access memory (RAM)	Oblik primarne računalne memorije čijem se sadržaju može izravno pristupiti	https://en.wikipedia.org/wiki/Random-access_memory
SPI	Sučelje za sinkronu serijsku komunikaciju	https://en.wikipedia.org/wiki/Serial_Peripheral_Interface
System on Chip (SoC)	Integrirani krug koji sadrži sve komponente računala ili elektroničkog sustava	https://en.wikipedia.org/wiki/System_on_a_chip
UART	Sučelje za asinkronu serijsku komunikaciju	https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter