

Univerza v Ljubljani

Fakulteta za elektrotehniko

David Zindović

EEG ROŽA - FIZIOLOŠKO VODENI ELEKTROMECHANSKI SISTEM

Diplomsko delo
Univerzitetni študijski program prve stopnje Elektrotehnika

Mentor: prof. dr. Gregor Geršak

Ljubljana, 2024

Št. teme: 00229 / 2023

Datum prijave: 13. 10. 2023

Univerza v Ljubljani, Fakulteta za elektrotehniko izdaja naslednjo nalogo:

Kandidat: **David Zindović**

Naslov: **EEG roža - fiziološko vodeni elektromehanski sistem**
EEG flower - physiologically controlled electromechanical system

Vrsta naloge: Diplomsko delo
Univerzitetni študijski program prve stopnje Elektrotehnika

Tematika naloge:

Preučite možnosti uporabe preprostih EEG signalov za vodenje večjega sistema. Izdelajte sistem, ki bo primeren za študiranje uporabnosti električne aktivnosti možganov ter bo uporaben tudi za promocijo znanosti in tehnike. Načrtujte, izdelajte in preskusite elektromehanski sistem, ki bo EEG voden. Za merilnik EEG signalov uporabite preprosto, cenovno ugodno nosljivo napravo z majhnim številom EEG elektrod.

Ljubljana, 8. 7. 2024

prof. dr. Gregor Geršak
Mentor

prof. dr. Janko Drnovšek
Predstojnik katedre



prof. dr. Marko Topič
dekan

Spodaj podpisani študent, David Zindovič, vpisna številka 64210323, avtor pisnega zaključnega dela študija z naslovom: EEG roža - fiziološko vodeni elektromehanski sistem,

IZJAVLJAM,

1. ¹ ☒ a) da je pisno zaključno delo študija rezultat mojega samostojnega dela;
b) da je pisno zaključno delo študija rezultat lastnega dela več kandidatov in izpolnjuje pogoje, ki jih Statut UL določa za skupna zaključna dela študija ter je v zahtevanem deležu rezultat mojega samostojnega dela;
2. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil;
3. da sem pri pripravi pisnega zaključnega dela študija ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
4. da soglašam z uporabo elektronske oblike pisnega zaključnega dela študija za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
5. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
6. da dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.
7. da dovoljujem uporabo mojega rojstnega datuma v zapisu COBISS.

V: Ljubljani
Datum: 8. 7. 2024

Podpis študenta:

Zindovič

¹ Obkrožite varianto a) ali b).

Univerza v Ljubljani

Fakulteta za elektrotehniko

David Zindović

EEG roža - fiziološko vodeni elektromehanski sistem

Diplomsko delo

Univerzitetni študijski program prve stopnje Elektrotehnika

Mentor: prof. dr. Gregor Geršak

Ljubljana, 2024

Zahvala

Sprva bi se rad zahvalil mentorju dr. Gregorju Geršaku za pozitivno energijo in strokovno podporo tekom projekta.

Zahvalil bi se rad Zavodu 404, ki mi je omogočil prostor in orodje za delo, ter tamkajšnjim kolegom za predloge in pomoč.

Zahvaljujem se tudi svoji družini za radovednost pri razvoju mojega projekta.

Posebej pa bi se rad zahvalil moji boljši polovici Moniki za podporo tekom študija in celotnega časa nastajanja izdelka.

Povzetek

V diplomskem delu pišem o izdelavi mehanizma, ki s pomočjo enega motorja omogoči sekvenčno rotacijo elementov, pri čemer je celoten mehanizem zasnovan tako, da spominja na rožo. Ključni sestavni deli izdelka so zobniki, ki omogočajo sočasno odpiranje rože (rotacijo sestavnih delov v obliki cvetov rože), medtem ko zamaknjeni delni zobniki omogočajo zaporedno rotacijo posameznih skupin cvetov. V svojem zaključnem delu prav tako opišem postopek načrtovanja in izdelave mehanizma ter možne izboljšave.

Celoten mehanizem je krmiljen v odvisnosti od meritev komercialnega EEG merilnika, natančneje v odvisnosti od stopnje umirjenosti uporabnika. Bolj kot je uporabnik umirjen, bolj se mehanska roža razcveti. Zaradi nedostopnosti Bluetooth modulov, ki jih priporoča proizvajalec, predstavlja sinhronizacija komunikacije med EEG merilnikom in mikrokrmilnikom oviro, rešitev katere opišem v programskem delu mojega zaključnega dela, kjer opišem med drugim tudi krmiljenje koračnega motorja s pomočjo gonilnika.

Za enostavnejšo uporabniško izkušnjo sem izdelal fizični uporabniški vmesnik, na katerem lahko uporabnik izbira med tremi možnostmi. Te so: neodvisno odpiranje in zapiranje mehanizma, beleženje meritev na računalniku ter odpiranje rože v odvisnosti od izmerjene stopnje umirjenosti. Poslednji dve možnosti je možno zagnati sočasno. Uporabniški vmesnik ima vgrajene tudi varnostne funkcije, ki omogočajo varno poseganje v obratovalno območje naprave za prekinitev obratovanja ali pa spremembo načina obratovanja s pomočjo tipke za potrditev, senzorja prisotnosti magneta vmesnika in časovne zakasnitve.

Ključne besede: elektroencefalogram, elektromehanska roža, EEG, zobniki, Bluetooth

Abstract

In my thesis, I write about the construction of a mechanism that, with the help of a single motor, enables sequential rotation of elements, whereby the entire mechanism is designed to resemble a flower. The key components of the product are gears, which enable simultaneous opening of the flower (rotation of components in the shape of flower petals), while rotated partial gears enable sequential rotation of individual groups of petals. In my concluding part, I also describe the process of designing and manufacturing the mechanism, as well as possible improvements.

The entire mechanism is controlled depending on the measurements of the commercial EEG measuring device, more precisely depending on the degree of calmness of the user. The more calm the user is, the more the mechanical flower blooms. Due to the unavailability of Bluetooth modules recommended by the manufacturer, the synchronization of communication between the EEG measuring device and the microcontroller is an obstacle, the solution of which I describe in the software part of my final thesis, where I describe, among other things, the control of the stepper motor with the help of a driver.

For a simpler user experience, I created a physical user interface on which the user can choose between three options. These are: opening and closing the mechanism independently, recording measurements on a computer and opening the flower depending on the measured degree of calmness. The last two options can be run simultaneously. The user interface also has built-in safety functions that allow safe intervention in the operating area of the device to interrupt operation or change the mode of operation by means of a confirmation key, sensor for the presence of an interface magnet and time delay.

Key words: electroencephalogram, electromechanical flower, EEG, gears, Blue-tooth

Vsebina

1	Uvod	1
1.1	Elektroencefalografija	1
2	Merilno orodje	5
3	Aktuator	7
3.1	Zobniški mehanizmi	8
3.2	Načrtovanje mehanizma	8
3.2.1	Pogonski sklop	9
3.2.2	Segment s tremi listi	10
3.2.3	Segment s petimi listi	12
3.2.4	Izgled izdelka	14
3.3	Orodje za izdelavo aktuatorja	15
3.4	Popravki tekom izdelave	15
3.4.1	Dodatni elementi	17
4	Strojna oprema	19

4.1	Plošča za uporabnika	21
5	Programska oprema	25
5.1	Prejem in obdelava podatkov	25
5.2	Krmiljenje koračnega motorja	27
5.3	Regulacija odprtosti rože	30
5.4	Uporabniški vmesnik	31
6	Rezultati	35
7	Zaključek	39
7.1	Možne izboljšave	39
	Literatura	41

Seznam slik

1.1	Postavitev elektrod po sistemu 10-20.	2
2.1	EEG merilnik Neurosky Mindwave Mobile 2.	5
3.1	Izgled končnega izdelka od spredaj.	7
3.2	Pogled na segmente končnega izdelka.	9
3.3	Gonilni sestav.	10
3.4	Prerez segmenta s tremi listi.	10
3.5	Reduktor za segment.	11
3.6	Zobniški obroč segmenta s petimi listi.	11
3.7	Segment s tremi listi - zaprt.	12
3.8	Segment s tremi listi - odprt.	12
3.9	Prerez segmenta s petimi listi.	12
3.10	Segment s petimi listi - zaprt.	13
3.11	Segment s petimi listi - odprt.	13
3.12	Lista za segment s petimi listi.	13
3.13	Izračun koordinatnih točk cveta.	14

3.14	Objemka za gred s premerom 6 mm.	17
3.15	Objemka za gred s premerom 8 mm.	17
4.1	Shema za EEG ščit za Arduino Uno.	20
4.2	Tiskano vezje EEG ščit za Arduino Uno.	20
4.3	Računalniška slika tiskanega vezja.	21
4.4	Uporabniški vmesnik.	22
4.5	Uporabniški vmesnik povezan na ščit mikrokrmilnika.	23
5.1	Sestava paketa podatkov EEG merilnika podjetja Neurosky.	26
6.1	Končni izdelek.	36
6.2	Uporabnik in izdelek.	37

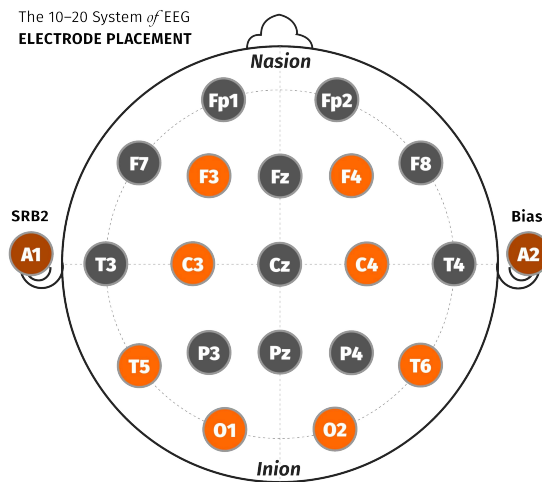
1 Uvod

Namen izdelave EEG rože je promocija tehnike in znanosti, saj projekt zajema merjenje in krmiljenje aktuatorja v obliki rože, ki je zanimiva navadnemu opazovalcu. S pomočjo EEG merilne metode sem želel predstaviti neinvazivno interakcijo med uporabnikom in napravo. Aktuator v obliki rože je krmiljen proporcionalno z umirjenostjo uporabnika, kar omogoča povratno zanko, saj se umirjenost uporabnika lahko spremeni ob pogledu na krmiljen aktuator, ki v primeru višje mere umirjenosti spominja na rožo, ki se je bolj razcvetela.

1.1 Elektroencefalografija

Pri elektroencefalografiji z elektrodami s površine glave odjemamo vsoto bioelektričnih aktivnosti celotnih možganov, torej skorje in vseh globljih struktur. Te “možganske valove”, ki jih generirajo številne možganske živčne celice in so bolj ali manj ritmična nihanja zunajceličnih tokov, zapisujemo z ustreznimi tehničnimi postopki. Zapis imenujemo elektroencefalogram (EEG) [1].

Za izvedbo meritev EEG je potrebna natančna postavitve elektrod na glavo merjenca. Najbolj razširjen sistem je 10/20, ki ga je razvil Herbert H. Jasper leta 1958. Postavil je 21 elektrod glede na možganski reženj, ki ga elektroda oz. sklop elektrod zaznava. Naziv sistema 10/20 se nanaša na delež loka med sprednjim in zadnjim delom možganov [2]. Zaradi potrebe po natančnejših meritvah so razvili postavitve z več elektrodami, kar je možno razbrati iz imena sistema. Z nižanjem druge številke v imenu je zmanjšan kot med elektrodami in s tem gostejša postavitve (npr. 10/10, 10/5) [3].



Slika 1.1: Postavitev elektrod po sistemu 10-20.

Podatki, pridobljeni preko EEG merilne metode, za enostavnejšo uporabo potrebujejo primerno obdelavo. Najenostavnejša analiza signalov z značilnimi frekvencami je Fourierova transformacija, ki nam kot izhodni podatek poda frekvenčni spekter. Omenjena merilna metoda nam kot podatek poda amplitude signalov različnih frekvenc možganskih valov, ki nam povedo več o aktivnosti možganov. Poznamo alfa, beta, gama, delta in theta valove, pri čemer bodo pri moji raziskavi najpomembnejši alfa in delta valovi, saj predstavljajo stanje umirjenosti in meditacije [4].

EEG je relativno enostavna, neinvazivna in cenena merilna metoda. Kljub omenjenim prednostim jo omejujejo vplivi mišičnih aktivnosti (predvsem obraznih mišic in mežikanja ter premikanja oči), ki jih zazna poleg možganske aktivnosti. Alternativni merilni tehnologiji brez omenjenih težav sta pozitronska emisijska tomografija (PET) in funkcionalna magnetna resonanca (fMRI). Pri PET opazujemo aktivnost posameznih delov možganov z merjenjem radioaktivnosti, ki se pojavi ob razpadu glukoze ob aktivnejšem delovanju posameznega dela možganov, medtem ko pri fMRI merimo spremembe v nasičenosti krvi s kisikom. Najpogosteje so pridobljeni podatki prikazani kot numerične vrednosti ali s pomočjo grafikonov. Slabost omenjenih tehnologij je, da nam ne podajata informacij v dejanskem času (npr. s pomočjo FMRI lahko pridobimo informacije vsakih pet sekund) in, da sta zelo dragi [5].

Čas je zelo dragocen, posebno ko nam ga primanjkuje v primeru obveznosti in časovnih zahtev, ki povzročajo stres. Le temu se je mogoče izogniti brez stroškov, in sicer s sproščanjem. Globoka sprostitiv je posebno fiziološko stanje, nasprotno tistemu, kar telo doživlja v stresu ali med paničnim napadom. To stanje je Herbert Benson leta 1975 opisal kot sprostilni odziv. Pri njem sodeluje niz fizioloških sprememb, med katerimi so: upočasnitev dihanja in srčnega utripa ter povečanje aktivnosti valov alfa v možganih. Za doseganje stanja globoke sproščenosti se najpogosteje uporabljajo metode, kot so preponsko dihanje, vodena vizualizacija, biološko-instrumentalna povratna zveza (biofeedback), pomirjujoča glasba ipd. [6].

2 Merilno orodje

Za izvedbo EEG meritev sem uporabil komercialni merilnik Mindwave Mobile 2, podjetja Neurosky. Razlog za izbor je bila dostopnost in enostavnost za uporabo. Merilnik je neinvaziven, saj ga je potrebno le nadeti na glavo, prisloniti merilno elektrodo na čelo in vpeti referenčno sponko na levo ušesno mečico (spodnji mejhejši del ušesa). Omenjeni EEG merilnik pošilja meritve preko Bluetooth protokola, pri čemer je mogoče pridobiti informacije o prisotnosti frekvenc možganskih valov, družin možganskih valov, kakovosti povezave in izbranih veličinah proizvajalca, kot sta mera umirjenosti in osredotočenosti v razponu od nič do sto [7].



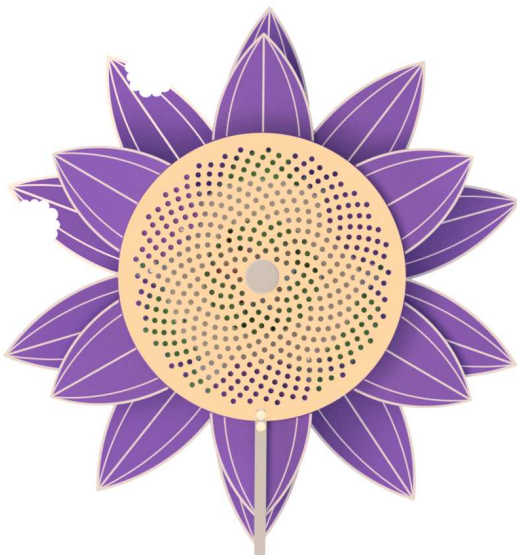
Slika 2.1: EEG merilnik Neurosky Mindwave Mobile 2.

Omenjeni merilnik meri vpliv možganskih valov približno na sredini čela oz. natančneje nekoliko bolj levo od sredine čela zaradi upogiba ročice, na koncu

katere se nahaja EEG merilna elektroda. To pomeni, da v primerjavi s sistemom 10-20, merimo zgolj v točki Fp1.

3 Aktuator

Aktuator je sestavljen iz več zobniških mehanizmov in pogonskega sestava, ki zagotavljata obračanje gredi, na katerih so leseni kosi v obliki listov. V obeh skrajnih legah aktuator spominja na rožo, zaprto ali odprto. Barve aktuatorja posnemajo barve španskega bezga. Da bodo uporabniki lahko sedeli in hkrati gledali v sredino aktuatorja (cveta), sem naredil podstavek z nastavljivo višino.



Slika 3.1: Izgled končnega izdelka od spredaj.

3.1 Zobniški mehanizmi

Zobnik je valjasto ali stožčasto telo, ki ima na obodu zobe. Njegova naloga je sprejem in prenos moči, prav tako pa ima zobnik velik razpon uporabnosti. So trpežni, zanesljivi, tihi, relativno preprosti, imajo visoko učinkovitost in možnost prenašanja velikih obremenitev. [8]

Prenos in sprejem moči, s tem tudi hitrosti vrtenja, lahko spreminjamo s spremembo zobniških razmerij. To je podatek, ki pove število obratov, ki jih bo naredil eden izmed zobnikov v eno smer, če jih drugi naredi toliko, kot je druga številka prestavnega razmerja v drugo smer. Prestavno razmerje je lahko tudi enako, torej ena na ena, kar pomeni da se vrtilna hitrost ohranja, vendar se zamenja smer vrtenja. Redukcija pri zobniških razmerjih pomeni nižanje vrtilne hitrosti (in s tem višanje navora), kar dosežemo z višjim prestavnim razmerjem, kot na primer pet proti ena. V tem primeru se bo moral prvi zobnik petkrat zavrteti v eno smer, da bo drugi zobnik naredil en obrat v drugo smer. Pričakovano je, da bo počasneje vrteč zobnik imel večjo površino oz. premer.

3.2 Načrtovanje mehanizma

Mehanizem znotraj EEG rože je sestavljen iz različnih sklopov, in sicer pogonskega sklopa ter dveh segmentov.

Moj cilj je bil sestaviti aktuator, ki bo potreboval le en motor za obračanje listov rože v radialni smeri sekvenčno. Prvotno sem načrtoval tri segmente, vendar sem tekom izvedbe sprejel le dva, in sicer zaradi nenačrtovanega povečanja velikosti, predvsem dolžine izdelka.

Izdelek sestavljajo 3D natisnjeni plastični zobniki in lasersko izrezani leseni zobniki ter aluminijasta palica s premerom šestih milimetrov uporabljena za pogonsko gred. Poleg omenjenih kosov so dodani tudi različni pritrdjevalni mehanizmi (vijaki in objemke), manjše gredi s premerom šest in osem milimetrov znotraj segmentov ter ležaji 608ZZ in 626ZZ.

Težavo pri prvotnem načrtovanju so predstavljale tolerance, saj leseni zobniki

niso imeli dobrega oprijema na aluminijaste palice, uporabljene za gredi, prav tako so pa tudi tanjše vezane plošče bile zelo ukrivljene. Za to je bilo potrebno celoten sestav stabilizirati v sledečih različicah.



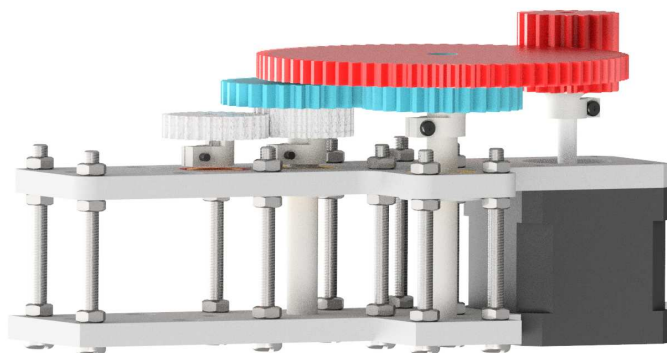
Slika 3.2: Pogled na segmente končnega izdelka.

3.2.1 Pogonski sklop

Pogonski sklop poganja celoten aktuator. Vsebuje koračni motor in reduktor z dodatnimi prenosi zaradi prostorskih omejitev izdelka.

Koračni motor FIT0278 proizvajalca DFROBOT potrebuje napajanje 5 V, kar je priročno, saj je možno z istim napajalnikom napajati tudi mikrokrmilnik. Motor upravlja gonilnik TMC2209.

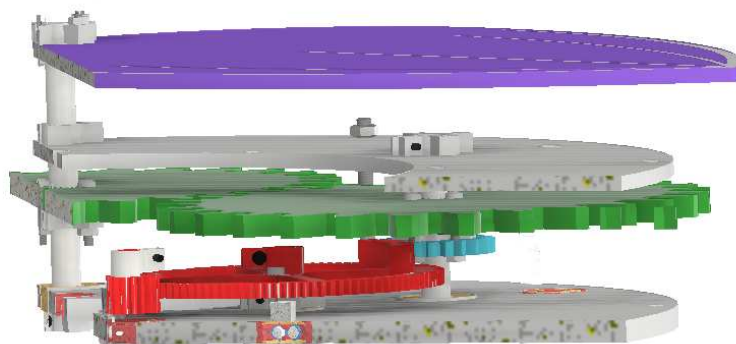
Izvor moči pogonskega sklopa je koračni motor, ki poganja reduktor z razmerjem pet proti ena. Sprejemni zobnik reduktorja ni možno neposredno povezati na gonilno gred mehanizma, saj bi s tem ustvarili pospeševalnik in obenem izničili učinek reduktorja. Zato so bili v pogonski sklop dodani trije zobniki, ki tvorijo skupaj razmerje prenosov ena proti ena. Prvi zobnik ima enako vrtilno hitrost kot večji zobnik reduktorja, nato pa ostala dva zobnika preneseta omenjeno vrtilno hitrost na gonilno gred, na katero je vpet skrajno levi zobnik na sliki 3.3 .



Slika 3.3: Gonilni sestav.

3.2.2 Segment s tremi listi

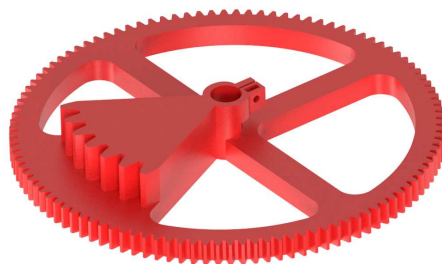
Segment s tremi listi sem načrtoval tako, da so na vsaki strani segmenta trije listi, skupaj šest listov na tem segmentu. Cilj segmenta je, da prejme vrtilno hitrost pogonske gredi in sekvenčno zavrti gredi listov na eni in zatem še na drugi strani.



Slika 3.4: Prerez segmenta s tremi listi.

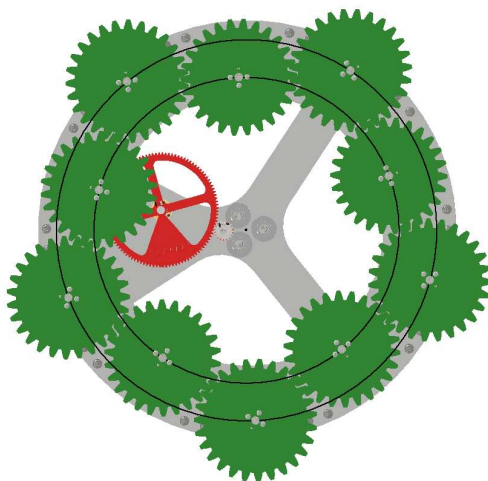
Vrtilno hitrost pogonske gredi prejme reduktor, ki nato preko so-osnega zobnika prenese moč na zobniški obroč. Mehanski zamik sem dosegel tako, da sem izdelal le šestino so-osnega zobnika in ga pri sestavi tovrstni reduktor primerno zamaknil pri vsakem segmentu. Vrtilno hitrost reduktorja prejme gonilni zobnik

za zobniški obroč, ki na sliki 3.4 obarvan modro. Vsak reduktor ima že vgrajeno objemko za gred, medtem ko ostali zobniki imajo štiri luknje namenjene pritrditvi na objemko, ki se jo nato pritrdi na gred.



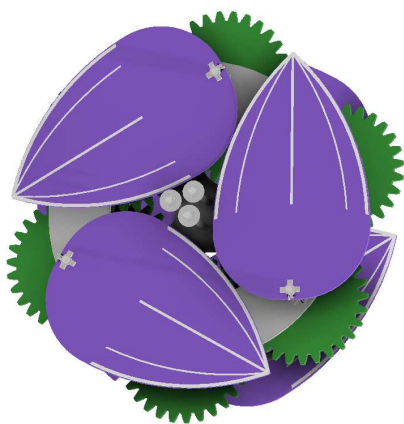
Slika 3.5: Reduktor za segment.

Zobniški obroč sestavljajo zobniki, ki vrtijo drug drugega, in so postavljeni v obliki kroga. Poskrbel sem za ustrezen zamik in s tem zagotovil, da se bodo odpirali le listi na zunanjem kolobarju zobniškega obroča.

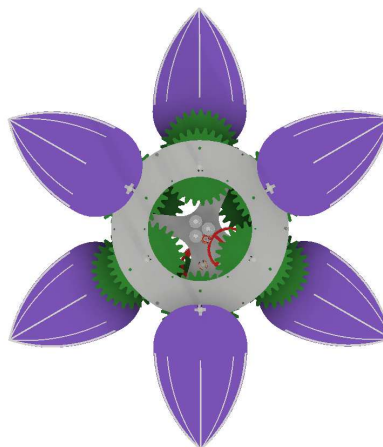


Slika 3.6: Zobniški obroč segmenta s petimi listi.

Zobnikov v zobniškem obroču je dvakrat toliko, kot je listov na posamezni strani segmenta. Vsaka stran segmenta ima svoj zobniški obroč.



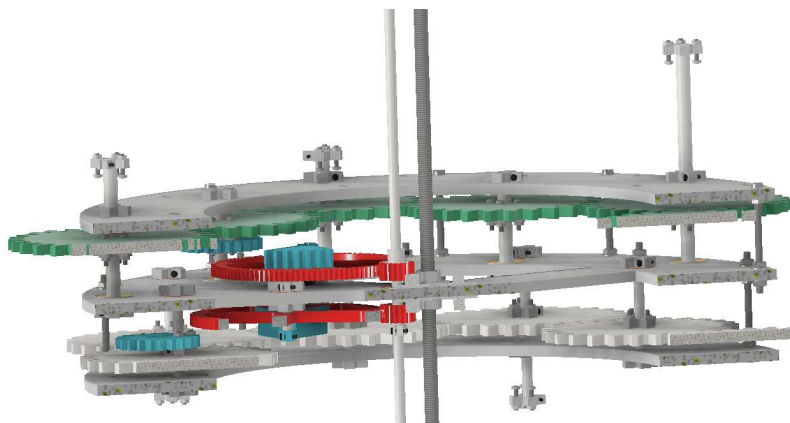
Slika 3.7: Segment s tremi listi
- zaprt.



Slika 3.8: Segment s tremi listi
- odprt.

3.2.3 Segment s petimi listi

Podobno kot segment s tremi listi je tudi segment s petimi listi načrtovan na enak način z razlikami v dimenzijah in številu listov. Posledično se ta segment razlikuje od prejšnjega tudi v številu zobnikov v zobniškem obroču in številu gredi.



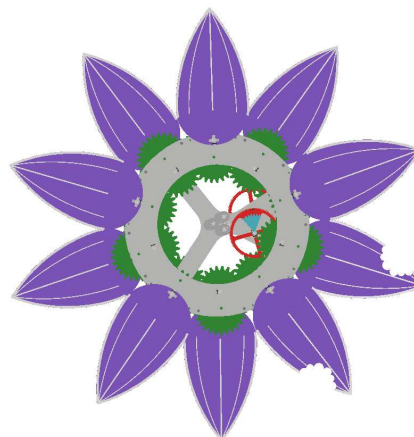
Slika 3.9: Prerez segmenta s petimi listi.

Razlog za razliko v dimenziji je ta, da bo oddaljenost listov tega segmenta od središča aktuatorja večja in se bodo s tem listi tega segmenta pojavili v ozadju

listov prejšnjega segmenta.

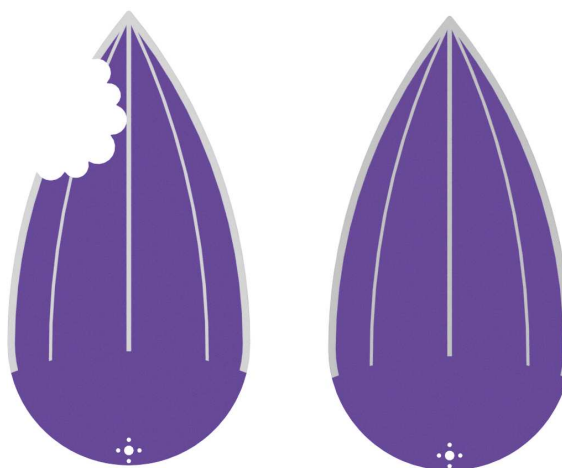


Slika 3.10: Segment s petimi listi - zaprt.



Slika 3.11: Segment s petimi listi - odprt.

Poleg tega so listi tega segmenta nekoliko ožji in daljši, prav tako se pa en list na vsaki strani razlikuje od ostalih s svojim "odgriznjениm delom", ki je potreben da se lahko list do konca pospravi. V kolikor odstranjenega dela ne bi bilo, lista ne bi mogel aktuator do konca zapreti, saj bi se zaletel v sosednjo gred.



Slika 3.12: Lista za segment s petimi listi.

3.2.4 Izgled izdelka

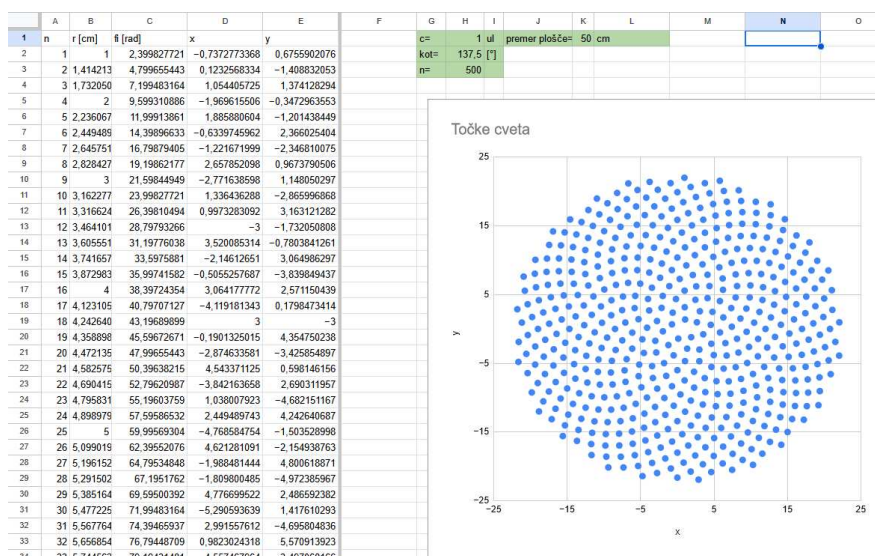
Pri izgledu sem se osredotočil na barvo cvetov, ki spominjajo na barvo cvetov španskega bezga, pri čemer sem dodal bele črte s pomočjo belega spreja in šablone. Zaradi manjšega razlitja barve je vsak cvet nekoliko drugačen.

Pri načrtovanju izgleda s sprednje strani sem se osredotočil na vzorec točk, ki jih vsebuje veliko cvetov. Ta vzorec je mogoče opisati s pomočjo Vogelovih formul (3.1) in (3.2), kjer je n zaporedno število točke (šteto ven iz središča), ϕ kot med prejšnjo in trenutno točko ter r razdalja med središčem in n -to točko [9].

$$\phi = n * 137,5^\circ \quad (3.1)$$

$$r = c\sqrt{n} \quad (3.2)$$

V ta namen sem izračunal koordinate točk s pomočjo formuliranja na platformi Google Sheets, pri čemer sem omogočil spreminjanje zgoraj naštetih parametrov (pri izračunih predstavlja n največji indeks točke, torej zgornjo mejo), prav tako sem pa tudi dodal omejitev velikosti površine s premerom plošče, kar razprši ali zbliža točke. Pridobljene koordinate sem nato uporabil za načrtanje lukenj v sprednji plošči izdelka.



Slika 3.13: Izračun koordinatnih točk cveta.

3.3 Orodje za izdelavo aktuatorja

Za izdelavo aktuatorja so bila uporabljena različna orodja, ki jih bom v nadaljevanju razdelil glede na vrsto materiala, ki sem jo z izbranim orodjem obdeloval.

Za izdelavo plastičnih kosov je bil uporabljen 3D tiskalnik, v mojem primeru Prusa MK3s, v kombinaciji s PLA filamentom. 3D modeli so bili načrtani v programu Autodesk Inventor Professional. Večine zobnikov nisem 3D tiskal zaradi dolgega časa izdelave, vendar verjamem, da bi 3D natisnjeni zobniki lahko bili precej stabilnejši in boljši.

Za izdelavo lesenih kosov je bil uporabljen laserski rezalnik Hypercut, s pomočjo katerega sem izrezal večje število zobnikov in nosilne ter ravnalne plošče segmentov, prav tako pa tudi sprednjo in zadnjo okrasno ploščo, pri čemer je bila za nosilni in okrasni plošči uporabljena vezana plošča debeline šestih milimetrov. Vsi ostali kosi so narejeni iz vezane plošče debeline štirih milimetrov.

Stojalo za aktuator je narejeno iz dveh plošč lasersko razrezane vezane plošče, enega kosa ročno žagane MDF plošče in dveh kosov ročno žaganega lesa. Stojalo in sam aktuator imata pritrjeni kvadratni cevi različnih dimenzij, ki omogočata razstavljanje izdelka na dva kosa in tudi nastavitev višine izdelka. Za izboljšanje stabilnosti izdelka sem totalno dve vrvi križno med najvišjim delom prve kvadratne cevi in najnižnim delom druge kvadratne cevi ter obratno.

Za obdelavo kovinskih kosov (gredi, navojnih palic in kvadratnih cevi) sem uporabil žago za kovino in vrtalni stroj.

3.4 Popravki tekom izdelave

Prvotna različica aktuatorja ni imela ravnalnih plošč, zato sem narezal vse zobnike namenjene prenosu ena na ena zobniškega kroga in temeljno ploščo segmenta s tremi s listi na vsaki strani. Za tem sem v ploščo postavil ležaje in narezal gredi s pomočjo ročne žage na dimenzijo, za katero sem sprva mislil, da bo ustrezala. Po natikanju zobnikov na gredi sem odkril, da bo izvedbo oviralo dejstvo, da vezane plošče niso popolnoma ravne. To težavo sem rešil tako, da sem pozorneje

izbiral primerno ploščo in s tem, da sem upošteval dejstvo, da ima debelejša plošča manjšo verjenost za upogibanje. Zato je temeljna plošča debelejša od ravnalnih. Omenjena rešitev ni bila popolna, zato sem moral dodati poravnane luknje na temeljno in ravnalno ploščo, pri čemer sem lahko z vijaki zagotovil poravnavo in kompenziral upognjenost ravnalnih plošč z dodatnim privijačenjem ali odvijanjem.

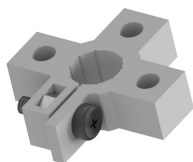
Tako sem uspel poravnati zobnike za obračanje gredi. Sprejemni del segmenta je predstavljal oviro, saj tistih kosov ni bilo mogoče pritrditi na ravnalno ploščo, tako da sem poravnavno zagotovil z dodajanjem objemk okoli ležaja. S tem sem temeljil na poravnanosti ležaja s temeljno ploščo.

Z dodajanjem ravnalne plošče sem moral prilagoditi dolžino gredi, na katerih se držijo listi. Gredi s premerom osmih milimetrov so predstavljale težavo s tem, da niso držale pozicije v ležaju za razliko od gredi s šest milimetrskim premerom. To, da so širše gredi padale ven iz pritrjenih ležajev, sem rešil z dodajanjem objemk na vsak stran ležaja.

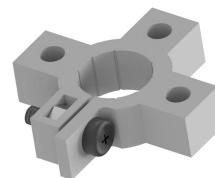
S pomočjo objemk sem uspel tudi pritrditi liste na gredi, ki sicer ne bi imeli dobrega oprijema na gredi in ne bi držali približno devetdeset stopinjski kot z gredjo. Za ta namen so bili dodani vijaki objemkam za držanje zobnika in objemke skupaj, poleg že obstoječega pritrjevalnega vijaka, s pomočjo vijačenja katerega lahko zmanjšam premer luknje objemke. Na enak način sem pritrdil tudi vsak zobniški obroč vsakega segmetna. Pri večjih zobnikih se je izkazal omenjeni način za pomankljivega, saj so časom začele objemke, in s tem zobniki, spodrsavati na gredeh. To pomeni, da čeprav se je zobniški krog obračal, se niso vsi listi, saj določeni zobniki niso obračali pristojne gredi. Omenjeno težavo sem rešil z dodajanjem sekundnega lepila med objemko in gredjo. Ker se je zobniški krog izkazal za neučinkovitega sem moral enega izmed zobnikov notranjega kolobarja umakniti. Ta isti zobnik se je izkazal kot koristen kasneje, pri servisiranju ostalih zobnikov (npr. pri nanašanju sekundnega lepila), saj je deloval kot zavora. Brez zavore bi moral po vsakem manjšem popravilu liste nastaviti na pravilno pozicijo in šele nato skleniti zobniški obroč.

3.4.1 Dodatni elementi

Tekom izdelave so se mnoge tolerance in načrti izkazali za neprimerne, zato sem želel težave rešiti na čim manj potraten način. Najbolj univerzalna rešitev za združitev elementov, nestacionarnost gredi in še nekaterih težav so bile 3D natisnjene objemke. Le te sem ustvaril v dveh dimenzijah: za gred s premerom šestih milimetrov in za gred s premerom osmih milimetrov. Križna oblika se upira ob najbližjo podlago, kar seveda povzroča nekaj trenja. Vsaki objemki sta dodani matica in vijak velikosti M2, s pomočjo vijačenja katerih je mogoče poskrbeti za zmanjšanje premera luknje zaradi česar objemka ostane na mestu in se ne premika po gredi.



Slika 3.14: Objemka za gred s premerom 6 mm.



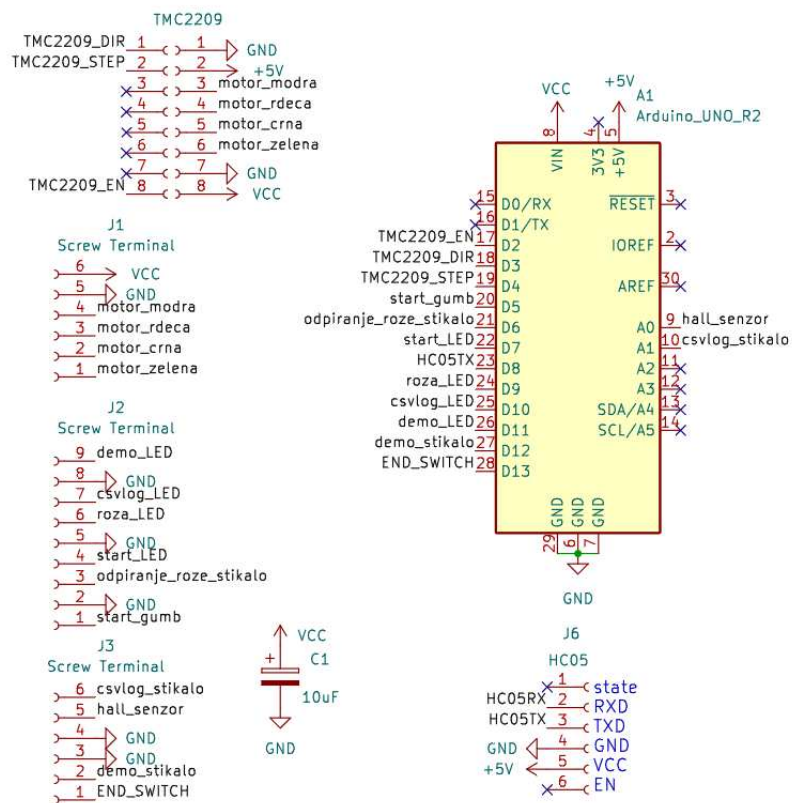
Slika 3.15: Objemka za gred s premerom 8 mm.

4 Strojna oprema

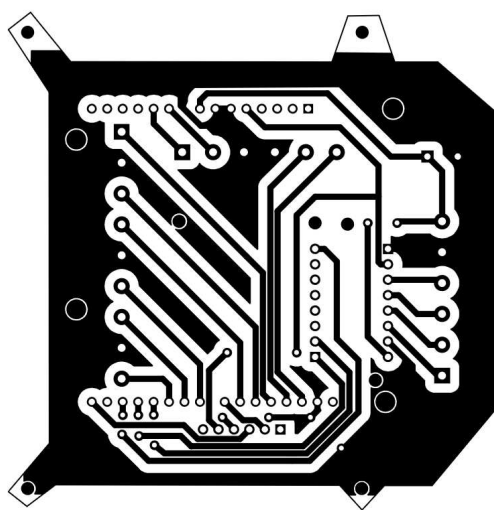
Elektronski sklop izdelka sestavljajo Arduino Uno mikrokrmilnik, TMC 2209 gonilnik, Bluetooth modul HC-05, Hallov senzor 49E, tipki, stikala, kondenzator in LED.

Arduino Uno mikrokrmilnik sem izbral zaradi enostavnega programiranja, izdelave t.i. ščita in montaže na zadnjo ploščo aktuatorja. Trije izhodi so namenjeni komunikaciji z gonilnikom za koračne motorje TMC2209, ki sem ga izbral, ker je bilo potrebno izbrani koračni motor FIT028 napajati s petimi volti. Ostali cenovno dostopnejši gonilniki so namenjeni uporabi z vsaj deset voltnimi koračnimi motorji. Preostanek izhodov je vezanih na uporabniški vmesnik preko vijačnih konektorjev (screw terminal). Za komunikacijo z Bluetooth modulom HC-05 je namenjen le en vhod mikrokrmilnika, saj le prejemamo podatke od modula in jih ne oddajamo. En analogen vhod je namenjen branju izhodne vrednosti Hallovega senzorja, medtem ko so ostali analogni priključki uporabljeni kot digitalni izhodi oz. vhodi ali pa so nepovezani.

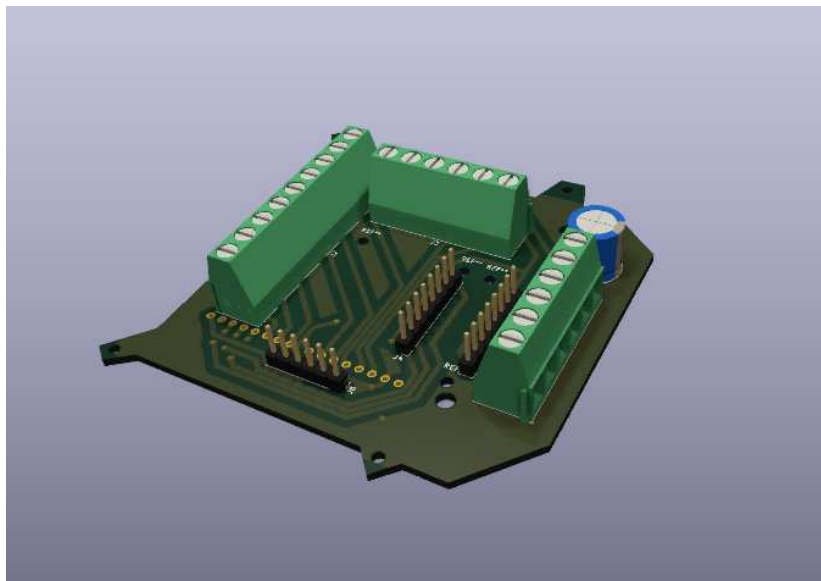
Za enostavnejšo vmestitev v izdelek sem izdelal tiskano vezje na CNC rezkarju Wegstr. Dodal sem nekaj dodatnih lukenj za pritrditev tiskanega vezja na zadnjo ploščo aktuatorja. Skrajno levi roki tiskanega vezja sta se izkazali za šibki, kar bi bilo potrebno spremeniti v prihodnosti. Dodal sem tudi luknji za pritrditev Bluetooth modula na ploščo s pomočjo laksa, saj ni prispajkan na vezje, vendar dan v ženske priključke. Razlog za to je potencialna potreba po ponovni konfiguraciji modula, zaradi česar bi ga bilo potrebno vzeti ven iz naprave.



Slika 4.1: Shema za EEG ščit za Arduino Uno.



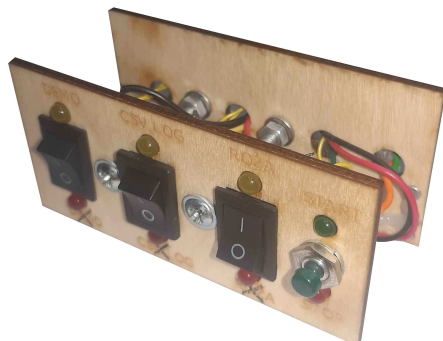
Slika 4.2: Tiskano vezje EEG ščit za Arduino Uno.



Slika 4.3: Računalniška slika tiskanega vezja.

4.1 Plošča za uporabnika

Dodatno sem naredil ploščo za uporabnika (v nadaljevanju UI plošča), ki obsega tri stikala, en gumb in osem LED diod ter vodnike, ki povezujejo UI ploščo s ščitom na mikrokrmilniku. Poleg tega ima plošča dva vijaka velikosti M4, ki držita sprednji in zadnji leseni plošči skupaj, ter magnet, zaradi katerega ostane na mestu, ko jo približamo ščitu za mikrokrmilnik. Ta ima prav tako pritrjen neodimijev magnet ob strani. Sama UI plošča ima pritrjen še en neodimijev magnet na drugem koncu zadnje strani, ki je namenjen signaliziranju prisotnosti in odsotnosti UI plošče Halloremu senzorju. V primeru prisotnosti odložene UI plošče ima omenjeni senzor magnetnega polja višje vrednosti na izhodu in nižje v primeru odsotnosti. Obseg omenjenih vrednosti je odvisen od polaritete magneta.



Slika 4.4: Uporabniški vmesnik.

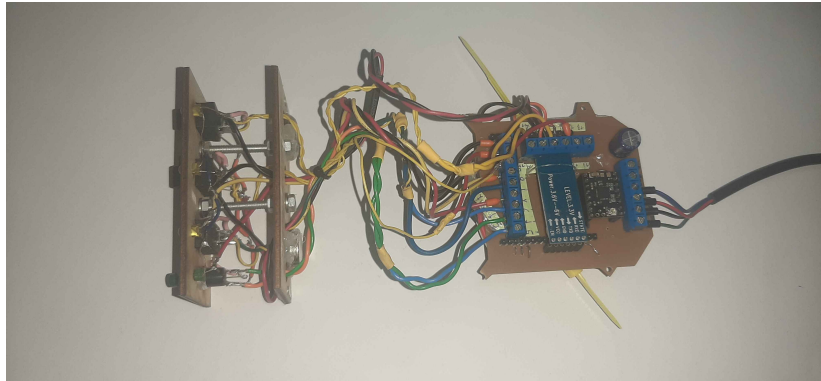
Plošča je namenjena preklopu med načini delovanja naprave. Uporabnik ima na izbiro DEMO način, ki ima višjo prioriteto od ostalih dveh, in pri katerem se roža le odpira in zapira, neodvisno od prisotnosti meritev.

Poleg omenjenga načina lahko uporabnik izbere tudi CSVLOG in ROŽA načina, pri čemer prvi označuje beleženje izmerjenih EEG podatkov in izpis podatkov na serijski izhod, preko katerega programska oprema na računalniku prejme podatek in ga shrani v tabelo tipa CSV. Omenjeno programsko opremo sem pripravil za namen beleženja meritev in izvedbe preizkusov, vendar je ne bom opisal v mojem zaključnem delu, saj je nisem še uporabil.

Z izbiro načina ROŽA pa sporočimo, da želimo izmerjene podatke EEG merilnika prikazati neposredno z odpiranjem in zapiranjem rože, proporcionalno s povprečnim nivojem umirjenosti uporabnika na podlagi zadnjih nekaj meritev.

Ob zagonu ima uporabnik zadosti časa umakniti se, nato se roža odpira, dokler se ne dotakne končnega stikala, ki naznani eno končno lego, nato pa se zapre tako da doseže drugo skrajno lego. Za drugo sem preko preizkušanja ugotovil, koliko korakov mora koračni motor narediti v obratno smer. Po rekaliibraciji ima uporabnik priložnost vzeti UI ploščo v roke in s pomočjo stikal nastaviti izbrani način, pri čemer LED diode prikazujejo trenutno izbrane načine. Končno odločitev potrdi s pritiskom na tipko START, pri čemer se prižge zelena LED dioda. Nato uporabnik ploščo odloži in ima čas, da se umakne izven območja obratovanja rože. Za tem naprava deluje v izbranem načinu, pri čemer lahko

uporabnik vmesna stanja razbere preko kombinacij prižganih LED diod na UI plošči, ki je vidna ob pogledu iz strani. Če želi uporabnik obratovanje prekiniti, lahko odstrani UI ploščo, kar zamrzne celotno napravo, izbere nov način in ga potrdi. Za tem se bo naprava rekalkibrirala (ponovno odprla in zaprla) in obratovala v izbranem načinu.



Slika 4.5: Uporabniški vmesnik povezan na ščit mikrokrmilnika.

5 Programska oprema

5.1 Prejem in obdelava podatkov

Podatke s strani EEG merilnika prejmemo preko Bluetooth komunikacijskega protokola, ki ga pokrije HC-05 modul in nam preko umetno ustvarjene serijske komunikacije, s pomočjo knjižnice AltSoftSerial, sporoči izmerjene podatke Arduino Uno mikrokrmilniku. Razlog za uporabo omenjene knjižnice je, da lahko teče serijska komunikacija med modulom in mikrokrmilnikom ter med mikrokrmilnikom in računalnikom, kar potrebujemo če želimo beležiti podatke na računalniku ali pa spremljati izpis za namen razhroščevanja. Proizvajalec merilnika Neurosky je podal na svoji spletni strani podroben vpogled v obdelavo prejetih podatkov oz. payload-a. Ker je merilnik Neurosky Mobile Mindwave 2 že nekaj časa na trgu, sem uspel najti nekaj knjižnic, ki optimizirajo nekatere nevšečnosti izvirnega programa proizvajalca.

Odločil sem se uporabiti knjižnico, ki jo je George Profenza ustvaril leta 2015. Prejemanje podatkov je enostavno, v kolikor je to edina stvar, ki jo mora mikrokrmilnik početi. Težava nastane, ko mora mikrokrmilnik več operacij opravljati s čim manjšim časovnim zamikom, saj je potrebno pravočasno prebrati podatke z Bluetooth modula. Za rešitev te ovire je potrebno upoštevati, da Mindwave Mobile 2 približno vsako sekundo pošlje nove podatke. S tem lahko program sestavimo tako, da ko mine vsaj ena sekunda, namenimo procesorski čas samo zanki za pridobitev zadnjih meritev. To storimo tako, da primerjamo prejšnjo vrednost umirjenosti z novo. Ko se nova posodobi, bo najverjetneje zajemala drugo vrednosti, saj je možnost za identično mero umirjenosti dokaj nizka.

```
1 void merjenja(bool ali_reguliram)
2 {
```

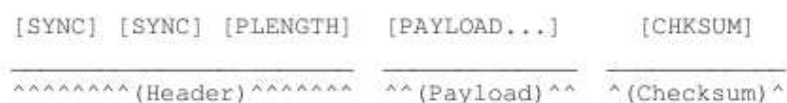


```

3  static bool ledica=0;
4  uint8_t eeg_prej=umirjenost;
5
6  if(blueetooth.available()>0){
7    digitalWrite(CSVLOG_LED,ledica);
8    ledica=!ledica;
9  }
10
11 while(umirjenost==eeg_prej){
12   mindwave.update(blueetooth,onMindwaveData);
13   digitalWrite(CSVLOG_LED,ledica);
14   ledica=!ledica;
15 }
16
17 if(ali_reguliram)regulacija(umir);
18 }

```

Iz prejetih podatkov je mogoče razbrati nivo umirjetnosti in osredotočenosti, kakovost povezave in prisotnost posameznih družin možganskih valov. Preden razberemo iz niza podatkov, to kar nas zanima, je potrebno najprej prebrati glavo prejetega paketa, ki vsebuje dva sinhronizacijska byta vrednosti 170. Tretji byte nam pa pove dolžino paketa, ki se seveda mora ujemati z našo prešteto dolžino. Za konec moramo tudi sešteti vse byte od četrtega do vključno predzadnjega in vsoto primerjati z zadnjim bytom, ki predstavlja checksum in je dodaten način preverjanja pravilnosti prejetega paketa [10].



Slika 5.1: Sestava paketa podatkov EEG merilnika podjetja Neurosky.

Delo z omenjenim EEG merilnikom je predstavljalo kar nekaj težav, saj proizvajalec nikjer ni delil navodil za delo z Bluetooth modulom HC-05. Čeprav je Neurosky delil s skupnostjo primer programa in s tem olajšal razumevanje protokola, je naprava mišljena za uporabo z BlueSmirf modulom, ki ga pa že dlje časa ni mogoče dobiti. Težave pri različnih uporabnikih na spletu so razmeroma podobne.

Največjo težavo je predstavljala sinhronizacija Bluetooth modula HC-05 z napravo Mindwave Mobile 2. Povezava obeh naprav je bila enostavna, prav tako nastavitve hitrosti izmenjave podatkov, vendar podatki dejansko niso prišli do Bluetooth modula. To težavo sem rešil tako, da sem po prižigu modula vklopil EEG merilnik in za tem se je vzpostavila avtomatska povezava. Po prvem utripu lučk na modulu, ki naznanijo povezano stanje, sem merilnik ponovno ugasnil. Za tem sem počakal približno tri do pet minut in povezavo ponovno vzpostavil. Za tem so podatki takoj prispeli na modul, kar sem videl na Tx priključku na modulu. Vzrok za tovrstno obanašanje ni jasen, vendar je omenjena rešitev ponovljiva in očitno deluje.

Poleg omenjene ovire s strojno opremo je veliko oviro predstavljala obdelava podatkov, predvsem kar se pravočasne izvedbe tiče. Sprva, ko sem uporabljal omenjeno knjižnico, je vse dobro delovalo, vendar sem kasneje ugotovil, da je bilo to zaradi tega, ker sem namenil osrednji funkciji knjižnice veliko procesorskega časa z malo časa med posodobitvami. Moja ideja je bila uporabiti ostale časovnike Arduino Uno mikrokrmilnika in namenjati procesorski čas s pomočjo knjižnice TimerOne, vendar je to s časom povzročilo svojevrstne težave. Na koncu se je izkazala kot primerna rešitev enostavna primerjava razlike trenutne vrednosti časovnika, ki teče v ozadju funkcije `micros()`, ki šteje minulo število mikrosekund od prvega klica, in spremenljivke, ki drži čas zadnje posodobitve. Ko je bila razlika vsaj milijon mikrosekund, sem namenil procesorski čas prejemanju in obdelavi meritev.

5.2 Krmiljenje koračnega motorja

Za krmiljenje koračnega motorja sem spisal funkcijo z imenom `vrtenje`, ki kot vhodne parametre sprejme hitrost vrtenja koračnega motorja, smer, število ponovitev v zanki in ciljno pozicijo za namen regulacije. Predzadnji vhodni parameter neposredno določa čas izvajanja funkcije `vrtenje` in obenem ločljivost korakov. Kot izhodni podatek funkcija vrne podatek, če je roža prišla do minimuma, kar sporoči proženje končnega stikala, do maksimuma, ki je programsko nastavljena številka korakov motorja, ali do dane ciljne pozicije.

Funkcija na podlagi vhodnega podatka hitrosti določi potreben čas do naslednjega proženja, pri čemer pošlje mikrokrmilnik pulz na priključek za korak (step pin) na gonilniku TMC2209. Poleg tega je seveda potrebno določiti tudi smer, kar zasede ločen priključek (direction pin), ter imeti možnost onemogočiti motor, torej ga ustaviti, pri čemer podatek za to pošljemo prav tako na ločen priključek (enable pin).

Ta funkcija je uporabljena tudi v funkciji za odpiranje rože oz. postavitev na začetno pozicijo in v funkciji za sprotno regulacijo odprtosti.

```
1 bool vrtenje(int hitrost, bool smer, int stevilo_korakov, int
    poz_cilj)
2 {
3     static int stepsPerSecond;
4     static int pozicija=MAX_POZICIJA_MOTORJA;
5     static uint32_t micros_prej=micros();
6     bool alarm=0;
7
8
9     if(poz_cilj==-1)
10    {
11        if(smer) stepsPerSecond = hitrost;
12        else stepsPerSecond = -hitrost;
13    }
14     else if (pozicija<poz_cilj) {stepsPerSecond = -hitrost;}
15     else stepsPerSecond = hitrost;
16
17     for(int i=stevilo_korakov;(i>0)&&(pozicija!=poz_cilj)&&(!alarm)
        ;--i)
18     {
19         if (((stepsPerSecond > 0) && (pozicija > 0))||((stepsPerSecond
            < 0) && (pozicija < MAX_POZICIJA_MOTORJA)))
20         {
21             static unsigned long nextChange = 0;
22             static uint8_t currentState = LOW;
23
24             if (stepsPerSecond == 0)
25             {
26                 currentState = LOW;
27                 digitalWrite(PIN_STEP1_STEP, LOW);}
28             else
29             {
```

```
30     if (micros() > nextChange)
31     {
32         if (currentState == LOW)
33         {
34             currentState = HIGH;
35             nextChange = micros() + 30;
36
37             if ((stepsPerSecond > 0) && digitalRead(END_SWITCH)&&(
                 pozicija>0)){pozicija--;}
38             else if ((stepsPerSecond < 0) &&(pozicija<
                 MAX_POZICIJA_MOTORJA)){pozicija++;}
39         }
40         else
41         {
42             currentState = LOW;
43             nextChange = micros() + (1000 * abs(1000.0f /
                 stepsPerSecond)) - 30;
44         }
45
46         if (stepsPerSecond > 0){digitalWrite(PIN_STEP1_DIRECTION,
            LOW);}
47         else{digitalWrite(PIN_STEP1_DIRECTION, HIGH);}
48
49         digitalWrite(PIN_STEP1_STEP, currentState);
50     }
51 }
52 }
53
54 if((!digitalRead(END_SWITCH))&&(stepsPerSecond>0))
55 {
56     alarm=1;
57     pozicija=0;
58 }
59 }
60 digitalWrite(PIN_STEP1_STEP, LOW);
61
62 return((pozicija>=MAX_POZICIJA_MOTORJA && stepsPerSecond<0) || (
    alarm) || (pozicija==poz_cilj));
63 }
```

5.3 Regulacija odprtosti rože

Za namen sprotnega reguliranja odprtosti oz. zaprtosti aktuatorja sem dodal funkcijo regulacija, ki kot vhodni argument sprejme trenutno mero umirjenosti in izračuna cilj, ki ga mora motor doseči na podlagi trenutne in prejšnje vrednosti. Osnovna ideja je bila, da ne želim preobremeniti aktuatorja s preveč nenadnim preskokom stanj in sem zato vpeljal odpiranje oz. zapiranje do aritmetične sredine zadnjih dveh vrednosti. Zaradi ovire v sinhronizaciji komunikacije omejena izvedba deluje zadovoljivo, vendar ne zelo hitro, saj je v primeru majhne spremembe med prejšnjim in trenutnim stanjem potrebno čakati približno eno sekundo, kljub majhnemu premiku aktuatorja. To sklepam, da bi bilo rešljivo z optimiziranjem časa prejema podatkov, saj dvomim da bi počasnejše obratovanje motorja znatno pripomoglo k prijetnejši uporabniški izkušnji.

Za namen regulacije je potrebno preslikati povprečno vrednost umirjenosti na podlagi zadnjih dveh meritev iz razpona 0 do 100 v razpon med 0 in največjim dovoljenim številom korakov motorja (v mojem primeru približno 29600). Za tem je potrebno dobljeno ciljno številko pozicije oz. nivoja odprtosti uporabiti pri klicu funkcije za krmiljenje motorja (vrtenje), ki bo vrnil logično enico, ko doseže ciljno vrednost ali ekstrem.

```
1 void regulacija(int trenutno_stanje)
2 {
3     static bool first_reg=false;
4     static int pozicija_cilj=1;
5     static int motor=0;
6     static int delta_motor=0;
7     int regulacijsko_povprecje;
8     static int meditacija_prej;
9
10    if(first_reg!=false || trenutno_stanje!=0)
11    {
12        if (first_reg==false)
13        {
14            if(trenutno_stanje!=meditacija_prej)meditacija_prej=
15                trenutno_stanje;
16            first_reg=!first_reg;
17        }
18        else
```

```
18 {
19     if(trenutno_stanje>meditacija_prej)
20     {
21         regulacijsko_povprecje=(trenutno_stanje-(trenutno_stanje-
22             meditacija_prej)/2);
23     }
24     else regulacijsko_povprecje=(trenutno_stanje+(-trenutno_stanje
25         +meditacija_prej)/2);
26     meditacija_prej=trenutno_stanje;
27 }
28 if(abs(regulacijsko_povprecje-motor)>MAX_SPREMEMBA) delta_motor=
29     MAX_SPREMEMBA*(regulacijsko_povprecje-motor)/abs(
30         regulacijsko_povprecje-motor);
31 else delta_motor=regulacijsko_povprecje-motor;
32
33 motor+=delta_motor;
34
35 if (motor>100) motor=100;
36 else if (motor<0) motor=0;
37
38 pozicija_cilj=map(motor,0,100,0,MAX_POZICIJA_MOTORJA);
39
40 while(!(vrtenje(HITROST_MOTORJA,0,KORAKI_MOTORJA,
41     MAX_POZICIJA_MOTORJA-pozicija_cilj))){}
42 }
```

5.4 Uporabniški vmesnik

Za namen uporabe uporabniškega vmesnika sta bili dodani funkcija za branje nastavljenе konfiguracije (ui config) in tudi pogosto branje vrednosti hallovega senzorja, s pomočjo katerega mikrokrmilnik izve, da je bil uporabniški vmesnik odstranjen od stalne pozicije, in zato ustavi program, ter po vrnitvi uporabniškega vmesnika odpre in zapre rožo z namenom rekaliibracije.

```
1 byte ui_config(void){
2
```

```
3 byte mask_log=0b010;
4 byte mask_roza=0b001;
5
6 bool roza_stikalo;
7 bool csvlog_stikalo;
8 bool demo_stikalo;
9 bool start_gumb=0;
10
11 while(!start_gumb)
12 {
13 start_gumb=!digitalRead(START_GUMB);
14 roza_stikalo=!digitalRead(ROZA_STIKALO);
15 csvlog_stikalo=!digitalRead(CSVLOG_STIKALO);
16 demo_stikalo=!digitalRead(DEMO_STIKALO);
17
18 digitalWrite(START_LED,start_gumb);
19 digitalWrite(ROZA_LED,roza_stikalo);
20 digitalWrite(CSVLOG_LED,csvlog_stikalo);
21 digitalWrite(DEMO_LED,demo_stikalo);
22 }
23
24 start_gumb=0;
25
26 #if CSVLOG
27 if((csvlog_stikalo<<1)&(mask_log))
28 {
29 Serial.println((csvlog_stikalo<<1)&(mask_log));
30 Serial.println((roza_stikalo<<0)&(mask_roza));
31 }
32 #endif
33
34 return (demo_stikalo<<2 | csvlog_stikalo<<1 | roza_stikalo<<0);
35 }
```

Ob prvem zagonu naprave mora biti uporabniški vmesnik na stacionarni poziciji, saj ob prižigu preberemo referenčno vrednost hallovega senzorja in s pomočjo ročno izbrane vrednosti odstopanja določimo ali je bil uporabniški vmesnik odstranjen.

```
1 void loop() {
2
3 static bool first_nastavitve=1;
```

```
4 static byte nastavitve;
5 static bool ponovna_nastavitev=0;
6
7 byte mask_demo=0b100;
8 byte mask_log=0b010;
9 byte mask_roza=0b001;
10
11 if(first_nastavitve)
12 {
13     nastavitve=ui_config();
14     first_nastavitve=!first_nastavitve;
15 }
16
17 while(analogRead(HALL_SENSOR)>=(HALL_SENSOR_AVRG+
    HALL_SENSOR_WIGGLE_ROOM))
18 {
19     if((nastavitve&mask_log)==2)
20     {
21         for(int c=0;c<100;c++)mindwave.update(bluetooth,onMindwaveData
            );
22     }
23
24     delay(SAFETY_CAKANJE);
25
26     if(ponovna_nastavitev)zapiiranje_roze();
27
28     digitalWrite(ROZA_LED,(nastavitve&mask_roza)==1);
29     digitalWrite(CSVLOG_LED,(nastavitve&mask_log)==2);
30     digitalWrite(DEMO_LED,(nastavitve&mask_demo)==4);
31     digitalWrite(START_LED,1);
32
33     delay(MODE_CHANGE_WAIT);
34
35     if(!ponovna_nastavitev)ponovna_nastavitev=1;
36
37     while((analogRead(HALL_SENSOR)<=(HALL_SENSOR_AVRG+
        HALL_SENSOR_WIGGLE_ROOM)) )
38     {
39         ObdelavaPodatkov((nastavitve&mask_demo)==4,(nastavitve&mask_log)
            ==2,(nastavitve&mask_roza)==1,ponovna_nastavitev);
40
41         if(ponovna_nastavitev)ponovna_nastavitev=0;
```



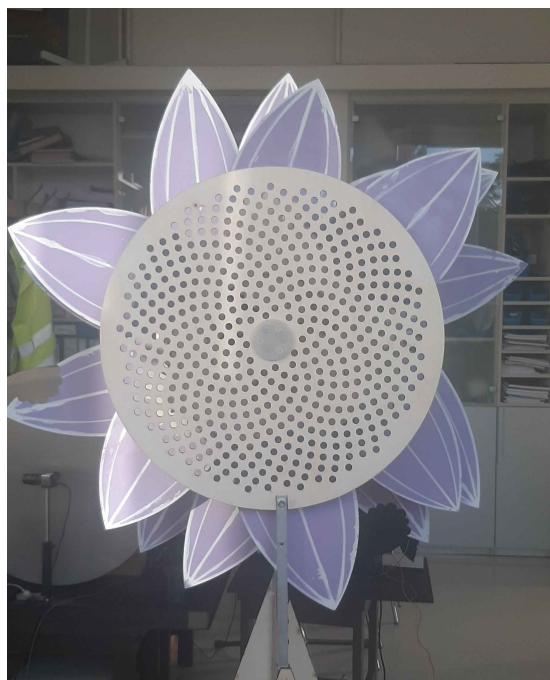
```
42  }
43
44  #if CSVLOG
45  if(((nastavitve&mask_log)<=3)&&((nastavitve&mask_log)>=1))
46  {
47    Serial.println("stop");
48  }
49  #endif
50
51  nastavitve=ui_config();
52  ponovna_nastavitev=1;
53
54  }
```

V kolikor bi želel uporabnik spremeniti delovanje naprave, bi bilo najlažje vzeti uporabniški vmesnik v roke in nastaviti nove nastavitve ter pritisniti tipko za zagon. V primeru odstranitve uporabniškega vmesnika se vse obstoječe meritve ponastavijo, roža se odpre in zapre po odložitvi UI plošče, v primeru beleženja podatkov se izpiše na serijski vmesnik 'stop' in na koncu se program vrne v funkcijo za branje konfiguracije uporabniškega vmesnika.

6 Rezultati

Končni izdelek predstavlja elektromehanski sistem, ki spominja na rožo. Mehanski sistem deluje dobro, vendar je slišen tekom obratovanja in rezultat odpiranja ali zapiranja je pogosto neponovljiv, zaradi preskakovanja sprejemnih zobnikov segmentov, spodrsavanja nekaterih gredi listov ali pa poškodovanja krhkih zobnikov kot posledica preskakovanja. Ugotovil sem, da sta najbolj kritični komponenti vsakega segmenta reduktor za segment (na sliki 3.5) in sprejemni zobnik zobniškega obroča (zobnik modre barve na sliki 3.4). Težava pri prvi komponenti je ta, da zaradi pritrjenosti le na spodnji strani lahko pride do nagibanja in posledično preskakovanja, medtem ko je težava pri drugi komponenti ta, da zaradi preskakovanja prve komponente lahko pride do odkrušenja zobov druge komponente. Posledično lahko poškodovan sprejemni zobnik zobniškega obroča povzroči nestabilno delovanje oz. prepreči rotacijo zobniškega obroča in s tem listov.

Zadovoljen sem z izgledom aktuatorja tako v zaprtem stanju, kot v razprtem. Za lepši izgled in boljšo stabilnost sem naredil stojalo, ki zelo pripomore k stabilnosti izdelka, čeprav se ta kljub vsemu nekoliko ziba med obratovanjem zaradi popuščanja napetosti vrvi.



Slika 6.1: Končni izdelek.

Izdelek sem preizkušal po delih tekom pisanja programske kode in tekom sestavljanja mehanskega sistema, zato sem pričakoval da bo izdelek deloval zadovoljivo kot celota. Mehanizem in uporabniški vmesnik sem preizkusil tako, da sem izdelek večkrat zagnal tako, da je ponovil odpiranje in zapiranje, pri čemer sem spremljal odstopanje med stanji popolnoma razcvetelega aktuatorja, ki se je izkazalo za zanemarljivo.

Končni preizkus je zajemal obratovanje mehanizma v odvisnosti od izmerjene vrednosti umirjenosti, pri čemer je izdelek obratoval primerljivo s preizkusom samega mehanizma. Opazil sem učinek povratne zanke, kjer se je aktuator bolj razcvetel v primeru pravilnih dejanj za doseganje višje mere umirjenosti. Predvidevam, da ko uporabnik enkrat ugotovi, katera dejanja so resnično sproščujoča, lahko zlahka popolnoma razcveti aktuator. Prav tako predvidevam, da bo ob prvem premiku aktuatorja opazen upad vrednosti umirjenosti zaradi motečih zvokov, zibanja aktuatorja in odpiranja aktuatorja, kar vse doprinese k manj umirjenemu počutju uporabnika.



Slika 6.2: Uporabnik in izdelek.

Tekom preizkušanja sem opazil, da obstaja možnost dodatne obrabe sestavnih delov v tem načinu obratovanja, kar bi lahko preprečil z zamenjavo reguliranega krmiljenja aktuatora z enkratnim odpiranjem. Omenjeni način bi približno pol minute ali eno minuto zajemal podatke in nato ustvaril povprečje, ki bi predstavljalo mero razcvetelosti aktuatorja, do katere bi se izdelek razcvetel za tem in uporabniku prikazal končni rezultat, za razliko od odpiranja in zapiranja v realnem času. Namesto tega sem se odločil, da bom izboljšal kritične sestavne dele in s tem ojačal sistem.

Ugotovil sem, da največjo oviro pri uporabi predstavljajo slabi kontakti na uporabniškem vmesniku in povezava EEG merilnika s strojno opremo naprave.

7 Zaključek

Končni rezultat moje diplomske naloge je zasnovan, načrtan in izdelan roži podobni elektromehanski aktuator, ki ga lahko uporabnik krmili (razcveti) s pomočjo merjenja možganske aktivnosti. Sistem s povratno zanko se je izkazal, da deluje, in da bi bil primeren za promocijo znanosti in tehnike.

7.1 Možne izboljšave

Izdelek bi lahko izboljšal predvsem v fizični dostopnosti kritičnih delov, saj bo v primeru poškodb teh delov večje popravilo zelo zahtevno, ker bo potrebno celoten izdelek razstaviti. Pri razstavljanju je potrebno biti pozoren na gonilno gred, saj so ležaji na njej zelo močno pritrjeni in je potrebno segmente na pazljiv način sneti z ležajev. Poleg tega bi stabiliziral reduktor za vsak segment in izdelal sprejemni zobnik za zobniški obroč iz 3D natisnjene plastike, kar predvidevam da bi znatno izboljšalo obratovanje izdelka.

Poleg samega mehanizma rože bi bile možne izboljšave tudi na spoju med stojalom in mehanizmom, saj je to trenutno najbolj mehansko šibek del izdelka.

Pri strojni opremi bi nadomestil Bluetooth modul HC-05 z modulom Blue-Smirf, ki ga tudi proizvajalec EEG merilne naprave Neurosky priporoča, ali pa drugim NRF modulom oz. NRF razvojno ploščo.

Literatura

- [1] L. Kolnik in V. Mir, “Grafoelementi možganske električne aktivnosti,” *Medicinski razgledi*, vol. 37, no. 2, str. 5–10, 1998. [Dostopano: 10. 12. 2023].
- [2] M. Seeck, L. Koessler, T. Bast, F. Leijten, C. Michel, C. Baumgartner, B. He in S. Beniczky, “The standardized eeg electrode array of the ifcn.,” *Clinical Neurophysiology*, vol. 128, no. 10, str. 2070–2077, 2017. [Dostopano: 12. 12. 2023].
- [3] Z. Koudelková in M. Strimska, “10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems,” *MATEC Web of Conferences*, vol. 34, no. 4, str. 1600–1611, 2007. [Dostopano: 12. 12. 2023].
- [4] Z. Koudelková in M. Strmiska, “Introduction to the identification of brain waves based on their frequency.” Dosegljivo: https://www.researchgate.net/publication/328097315_Introduction_to_the_identification_of_brain_waves_based_on_their_frequency. [Dostopano: 12. 6. 2013].
- [5] N. Jaušovec, “Nadarjeni možgani,” *Anthropos (Ljubljana)*, vol. 4, no. 6, str. 73–88, 1929. [Dostopano: 12. 12. 2023].
- [6] E. J. Bourne, *Anksioznost in fobije: priročnik za samopomoč: [s praktičnimi nasveti za sproščanje, telesno vadbo, premagovanje panike, izpostavljanje, odpravljanje zmotnih prepričanj in specifičnih fobij, asertivnost, samospoštovanje, prehrano, uporabo zdravil, meditacijo]*. Ljubljana: Modrijan, 2014.
- [7] “Mindwave mobile 2 brainwave sensing headset.” Dosegljivo: <https://store.neurosky.com/pages/mindwave>. [Dostopano: 4. 6. 2024].

- [8] I. Janežič, *Strojni elementi 2*. Ljubljana: Tehniška založba Slovenije, 2001.
- [9] “Phyllotaxis.” Dosegljivo: <https://www.algorithmicbotany.org/papers/abop/abop-ch4.pdf>. [Dostopano: 4. 7. 2024].
- [10] “Mindset communications protocol.” Dosegljivo: https://developer.neurosky.com/docs/lib/exe/fetch.php?media=mindset_communications_protocol.pdf. [Dostopano: 8. 6. 2024].