

ŠOLSKI CENTER VELENJE  
ELEKTRO IN RAČUNALNIŠKA ŠOLA

Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ SAŠA REGIJE

RAZISKOVALNA NALOGA

# **PAMETNA PLEZALNA DESKA**

**Ali lahko nadgradiš svojo moč z inteligentno plezalno desko**

Tematsko področje: ELEKTROTEHNIKA, ELEKTRONIKA in ROBOTIKA

Avtorja:

Lovro Kočevar Ribič

Jaka Ledinek

Mentor:

Peter Vrčkovnik, dipl. ing. elektrotehnike

Velenje, 2026

Raziskovalna naloga je bila opravljena na Šolskem centru Velenje, Elektro in računalniški šoli.

Mentor: Peter Vrčkovnik, dipl. ing. elektrotehnik

Datum predstavitve: marec 2026

**KJUČNA DOKUMENTACIJSKA INFORMACIJA**

ŠD	ŠC Velenje – Elektro in računalniška šola, šolsko leto 2025/2026
KG	plezanje / trening / elektronika / povezljivost / upravljanje na daljavo
AV	KOČEVAR RIBIČ Lovro, LEDINEK Jaka
SA	VRČKOVNIK Peter
KZ	3320 Velenje, SLO, Trg mladosti 3
ZA	Šolski center Velenje
LI	2026
IN	Pametna plezalna deska
TD	Raziskovalna naloga
OP	
IJ	SL
Jl	sl/ en
AI	<p>Pametne plezalne deske so predvsem drage in velike, kar lahko manjšim plezalnim klubom predstavlja breme in se raje odločajo za standardne plezalne stene. Najina plezalna deska bi bila majhna in cenovno ugodna. Omogočala bi veliko funkcij, ki jih cenovno manj ugodne nimajo, oziroma jih nisva zasledila na trgu. Plošča bo omogočala dober napredek plezalcu in tudi klubu ter posamezniku zmanjšala strošek za nakup bolj napredne tehnologije.</p>

## KEYWORD DOCUMENTATION

ND	ŠC Velenje – Elektro in računalniška šola, šolsko leto 2024/2025
CX	climbing / training / electronics / connectivity / remote control.
AU	KOČEVAR RIBIČ Lovro, LEDINEK Jaka
AA	VRČKOVNIK Peter
PP	3320 Velenje, SLO, Trg mladosti 3
PB	School center Velenje
PY	2025
TI	Pametna plezalna deska
DT	RESEARCH WORK
NO	
LA	SL
AL	sl/ en
AI	Smart climbing boards are mostly expensive and large, which can be a burden for smaller climbing clubs, who prefer to opt for standard climbing walls. Our climbing board would be small and more affordable than others, but it will provide many features that are less affordable and some that we have not yet seen on the market. The board will allow good progress for the climber and also reduce the cost for the club or individual to purchase a little more advanced technology.

## KAZALO VSEBINE

1.	UVOD.....	1
2.	PONUDBA PAMETNIH SISTEMOV NA TRGU .....	2
2.1	Plezalna deska podjetja Climbro .....	3
2.2	Smartboard climbing .....	4
2.3	Moonboard .....	5
2.4	So ill connect board.....	6
2.5	Zaključek pregleda ponudbe.....	7
2.6	HIPOTEZE .....	8
3	MATERIALI IN METODE .....	9
3.1.	Idejna zasnova modela .....	9
3.2	Plezalna deska.....	11
3.3	Barvno markiranje oprimkov.....	12
3.4	IR senzor.....	14
3.5	LCD .....	15
3.6	Meritev sile .....	17
3.7	Princip delovanja merilnika sile .....	17
3.8	Ojačitev in obdelava signala.....	17
3.9	Umerjanje senzorja .....	18
3.10	Porabnost pri treningu .....	18
3.11	Krmilni sistem .....	20
3.12	Program .....	25
3.13	Mobilna aplikacija .....	27
3.14	Povezljivost .....	28
4	Končni model .....	30
5	Rezultati.....	32
6	FINANČNO VREDNOTENJE MODELA .....	33
7	PreIZkus s plezalci.....	34
8	Razprava .....	35

9	Zaključek .....	36
10.	VIRI IN LITERATURA.....	38
11.	ZAHVALA.....	39
12.	PRILOGE .....	40
12.1.	Program za esp34 .....	40
12.1.A.	Priloga A - main.cpp.....	40
12.2.	Program za strežnik.....	55
12.2.B.	Priloga B - add_program.html .....	55
12.2.C.	Priloga C - get_wrkt.php.....	56
12.2.D.	Priloga D - save_prog.php .....	57
12.1.E.	Priloga E - add_prog_sy.php.....	61
12.2.F.	Priloga F - db.php.....	64
12.2.G.	Priloga G - Struktura SQL baze podatokv .....	65

## KAZALO SLIK

slika 1: Plezalna deska podjetja Climbro.....	3
slika 2: Plezalna deska podjetja SmartBoard Climbing.....	4
slika 3: Plezalna stena podjetja MoonBoard.....	5
slika 4: Plezalna stena podjetja ILL Connect Borad.....	6
slika 5: Idejni model plezalne deske.....	9
slika 6: Plezalna deska.....	10
slika 8: Povezovanje modulov in tiskanih vezji. ....	12
slika 9: IR senzor razdalje .....	14
Slika 10: 20 x 16 I2C LCD prikazovalnik.....	15
slika 11: Meni za izbiranje programov na LCD prikazovalniki .....	16
slika 12: Prikaz parametrov na LCD prikazovalniku med izvajanjem programa.....	16
slika 13: LCD prikazovalnik ob koncu programa .....	16
slika 14: Merilnik sile.....	17
slika 15: Ojačevalnik za merjenje sile .....	18
slika 16: Krmiljenje RGB LED diod .....	20
slika 17: Vhodni pomični register 74HC165 in pripadajoče povezave.....	20
slika 18: Pretvornik s 3,3V na 5V logični signal s BSS138 .....	21
slika 19: Spodnji del vezja za krmiljenje RGB diod in IR senzorjev .....	23
slika 20: Pretvornik z 3,3V na 5V logični signal z BC817-16 in 74HC14.....	23
slika 21: Tiskano vezje za vhode in izhode, zgornja stran .....	24
slika 22: Povezava zgornjega in spodnjega tiskanega vezja za vhode in izhode .....	24
slika 23: Diagram poteka prvega programa. Prva verzija programa je bila napisana po tem diagramu poteka. ....	26
Slika 24: Spletna stran za dodajanje programov. ....	27

# 1 UVOD

Trenutno se večina plezalcev tako profesionalnih kot tistih, ki plezajo v prostem času, **vozi na treninge tudi po več 10 km**, nekateri pa so zaradi prilagajanja treningom primorani **zamenjati lokacijo bivališča**. To je zelo **neučinkovito** tako iz časovnega, ekološkega in finančnega vidika.

V tej raziskovalni nalogi bomo raziskovali in poizkušali, kako rešiti te probleme z uporabo **modernih tehnologij**.

Ker je glavna omejitev za treniranje plezanja doma prostor, je eden izmed ciljev **narediti napravo, ki omogoča simulacijo plezanja v čim manjšem prostoru** brez potrebe po dolgih vožnjah.

Drugi izziv pa je nadzor trenerja. Tukaj se vprašamo »**Kako bi lahko trenerja spravili do plezalce?**«. Odgovor na to vprašanje je **tehnologija**, ki nam to omogoča. Iz tega izvira drugi cilj, da trener **pripravi trening na daljavo in ga na daljavo tudi spremlja**.

Tretji problem pa je **cena**. Cilj raziskovalne naloge je poiskati rešitev za zgornja dva problema, ki si jo lahko **privoščijo plezalni klubi, športna društva in širša populacija**.

Pri izdelavi raziskovalne naloge smo se povezali s trenerjem slovenske plezalne reprezentance Gorazdom Hrenom, ki nam bo pomagal pri specifičnih potrebah za pripravo učinkovitega treninga in opreme.

Pomembno je, da v raziskovalni nalogi poiščemo način, kako **trenirati plezanje iz lastne hiše ali stanovanja z nadzorom trenerja na daljavo**. To bo pripomoglo k boljšem izkoristku prostega časa, treniranju iz udobja lastnega doma in zmanjšanju stresa zaradi dolgih voženj in gostega prometa.

## **2 PONUDBA PAMETNIH SISTEMOV NA TRGU**

Trenutno je na trgu veliko majhnih plošč za treniranje plezanja iz različnih materialov. Najpogostejši material je les, predvsem zaradi cene, lahkega obdelovanja in vzdržljivosti.

Na trgu je tudi veliko plošč za treniranje plezanja iz različnih plastik. Plastične plošče so manj odporne na udarce, vendar lahko ob pravih postopkih končne obdelave omogočajo boljši oprijem.

Na trgu sva zasledila kar nekaj modernejših plošč, ampak ni bila nobena izmed njih takšna, kakršno sva izdelala midva. Plošče, ki sva jih zasledila na trgu, niso omogočale merjenja sile iz leve in desne strani. Na trgu pa tudi nisva našla takšne, ki bi plezalcu povedala, kateri oprimek mora prijeti naslednje.

Pred začetkom raziskave sva pregledala tržišče. Na trgu sva našla naslednje plošče.

## 2.1 Plezalna deska podjetja Climbro

Prva izmed najdenih pametnih plošč je pametni hangboard po imenu Climbro, ki s pomočjo vgrajenih senzorjev meri moč prstov, vzdržljivost in napredek, ki ga naredi uporabnik. Deluje s pomočjo njihove pametne aplikacije po imenu Climbro, ki s pomočjo kvalitetno razvitega AI sistema pripravi trening, ki je personaliziran in sprejema ter daje rezultate oziroma informacije.



slika 1: Plezalna deska podjetja Climbro

Vir: <https://climbro.com/>

Njihova največja posebnost je, da aplikacija vsebuje poseben model umetne inteligence, ki zelo dobro pomaga uporabniku. Cena njihove plezalne plošče in aplikacije je 749.00 €. Spletna stran izdelka - <https://climbro.com>

## 2.2 Smartboard climbing

Naslednje kar sva zasledila, je sistem SmartBoard, to so merilni senzorji, ki jih namestiš na poljubno pametno ploščo. V podjetju SmartBoard so razvili aplikacijo SmartBoard Climbing, ki omogoča testiranje, personalizacijo vaje in analizo ter primerjavo podatkov. Pri SmartBoard Climbing ponujajo plezalno desko, ki je opremljena s senzorjema za merjenje moči. Smartboard pa je poleg svoje plezalne plošče razvil tudi aplikacijo, ki daje plezalcu takojšni povratek informacij njegovega plezanja. S pomočjo aplikacije lahko plezalec zelo natančno nadzoruje svoj napredek, cena njihove plošče je od 662,50 € do 2.590,00 €.

Spletna stran izdelka - SmartBoard Climbing, <https://www.smartboard-climbing.com/>



*slika 2: Plezalna deska podjetja  
SmartBoard Climbing Vir:  
<https://www.smartboard-climbing.com/>*

## 2.3 Moonboard

To je pa interaktivna plezalna stena z vsebovanimi LED lučkami, katere trener ali plezalec lahko poljubno nastavi, da potem pleza po tistih oprimkih, katere je prej nastavil. Plošča omogoča zelo veliko kreativnosti in omogoča plezalcu zelo dober trening. Na to ploščo se poveže preko bluetooth povezave. Pri podjetju MoonBoard so tudi naredili zelo dobro zasnovano aplikacijo,, v kateri opravljaš svojo pametno plezalno steno in si v njej lahko tudi narediš svoj račun in si shraniš najljubše kombinacije, da jih lahko tudi kdaj drugič preplezaš. Cena njihove plezalne stene z določenimi dodatki je 4550 €.

Njihova spletna stran: Using Your MoonBoard App, <https://moonclimbing.com/using-moonboard-app>



slika 3: Plezalna stena podjetja MoonBoard  
Vir: <https://en.androidguias.com/climbing-apps/>

## 2.4 So ill connect board

ILL Connect Board je plezalni sistem trening plošče, ki z LED lučkami in aplikacijo, ki omogoča iskanje problemov, beleženje vzponov in tudi povezovanje z drugimi plezalci. Je večja plezalna stena, ki je opremljena z oprimki, ki se zasvetijo po tisti poti, po kateri mora plezalec plezati, svoje poti lahko plezalci objavljajo na njihovi aplikaciji, kjer lahko tekmujejo med sabo, v kakšnem času splezajo določeno pot. Pri soillholds imajo tudi manjše plezalne deske, a te niso opremljene z nobeno tehnologijo. Cene njihovih izdelkov so okoli 10.000 €.

Njihova spletna stran-soillholds.com, <https://soillholds.com/>



slika 4: Plezalna stena podjetja ILL Connect Borad

Vir: [https://soillholds.com/products/connect-board?srsId=AfmBOor8Cufgqyb2-uKOUdJKZ5aW\\_XEwaYOi082EHJN2fX980jxmN3J\\_&variant=49256600142100](https://soillholds.com/products/connect-board?srsId=AfmBOor8Cufgqyb2-uKOUdJKZ5aW_XEwaYOi082EHJN2fX980jxmN3J_&variant=49256600142100)

## 2.5 Zaključek pregleda ponudbe

Glede na ponudbo ponudnikov lahko razberemo, da podjetja ponujajo zelo zanimive funkcije, ampak so njihove plezalne plošče težje dostopne. Najina plezalna deska bo cenovno bolj dostopna kot ostale. Na trgu zaenkrat ni take plošče, ki bi omogočala merjenje sile na levi in desni roki, merjenje hitrosti in sledenje algoritmu med plezanjem.

Najina plezalna deska bi predvsem odpravila težavo tega, da so določene plezalne plošče zelo velike ali pa da nimajo vseh funkcij, ki jih najina omogoča.

Najina plezalna deska je tudi predvsem cenovno ugodnejša kot plošče drugih ponudnikov. Takšne plošče, ki je tako mala in omogoča toliko funkcij še ni na trgu in verjameva, da bi lahko bila najina deska zelo popularna med plezalci.

## 2.6 HIPOTEZE

Pred začetkom raziskovanja sva si zastavila naslednje hipoteze:

**HIPOTEZA I:** Plezalna deska bo primerna za montažo v stanovanjske prostore zaradi majhne velikosti.

**TESTIRANJE HIPOTEZE I:** Hipotezo sva testirala s primerjanjem velikosti plezalne plošče s povprečnimi velikostmi stanovanjskih prostorov. Najina plezalna deska se lahko montira v večino stanovanj.

**HIPOTEZA II:** Plezalna deska bo omogočala trening na daljavo.

**TESTIRANJE HIPOTEZE II:** Hipotezo sva testirala tako, da sva pripravila trening na mobilni aplikaciji in ga poslala na krmilni modul na plezalni deski.

**HIPOTEZA III:** Plezalna deska bo omogočala merjenje dviga plezalca, sledenju oprimkov in merjenju sile na levi in desni plošči.

**TESTIRANJE HIPOTEZE III:** Hipotezo sva testirala tako, da sva na plezalno ploščo namestila ultrazvočni senzor, kateri bo meril dvig plezalca in vrnil meritve višine dviga, čas med minimalno in maksimalno točko dviga. Plezalec je naredil vse skupaj 10 dvigov in sva tako dobila meritve.

**HIPOTEZA IV:** Plezalna deska bo omogočala učinkovito izvajanje treningov.

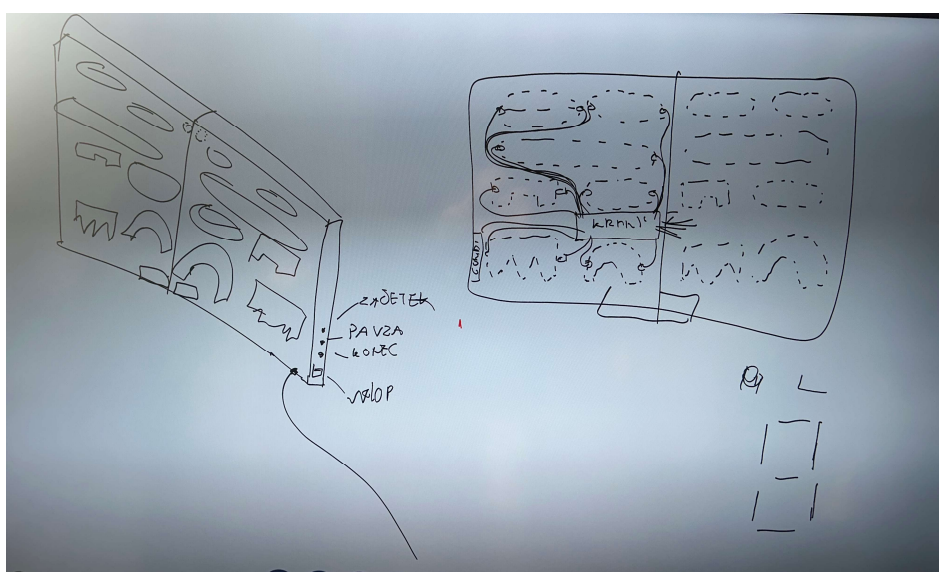
**TESTIRANJE HIPOTEZE IV:** Hipotezo sva preverila tako, da sva plezalno dasko dala na testiranje trenerju slovenske plezalne reprezentance Gorazdu Hrenu, ki pa je testiral plezalno desko in pridobil rezultate treningov.

## 3 MATERIALI IN METODE

### 3.1 Idejna zasnova modela

Najina ideja je bila, da bi lahko olajšala trening plezalcem s tem, da bi jim naredila pametno plezalno desko, ki bo plezalcem ponazorila, kje se morejo prijeti.

Deska bo trenerjem plezalcem in tudi plezalcu samemu pokazala različne informacije (sila in hitrost dviga). Deska bo imela različne figure, katerim bo plezalec sledil in s tem treniral in izboljševal svojo tehniko.



slika 5: Idejni model plezalne deske

Vir: Lasten

Deska bo podobna kot tista, ki je pripeta spodaj, vendar bo prerezana na polovico in v luknjah za prijemanje oz. držanje bodo LED lučke, ki bo seznanjala plezalca, kje je naslednji oprijem. Na sredini plošče bo nameščen ultrazvočni senzor, s katerim bomo lahko merili hitrost dviga plezalca. Plošča bo tudi vsebovala merilni sile, njegova naloga pa bo, da bo meril kolikšno silo se bo plezalec dvignil. Poleg deske bova naredila aplikacijo, ki bo trenerju omogočala izbiro figure. Trener bo s pomočjo aplikacije izbral zeleno število oprimkov, v katerih se bodo zaporedno prižigale led lučke. Kombinacijo bo moral s preprijemanjem narediti plezalec. Na aplikaciji bo po koncu treninga vidna reakcija plezalca, razporeditev moči na levi in desni deski in hitrost dviga.

Najino plezalno desko bodo sestavljale naslednje komponente:

- lesena plezalna deska (leva in desna plezalna plošča)
- krmilna enota
- LCD prikazovalnik
- integrirane RGB svetleče diode
- IR senzor
- merilnik sile
- mobilna aplikacija

Primer plezalne deske:



*slika 6: Plezalna deska*

*Vir: <https://trenirajznaravo.com/trening/>*

## 3.2 Plezalna deska

Zaradi lažje in cenejše prototipne izdelave sva se odločila, da bo deska narejena iz lesa. Razdeljena je na dva dela, ki bosta na sredini povezana z vodilom. Vsaka stran je prek merilnika sile pritrjena na steno. To omogoča natančne meritve sile na posamezni roki.

Na sprednji strani deske so utori, za katere se primemo in izvajamo vaje. V vsakem utoru je RGB LED dioda, ki pove status oprimka. S strani pa je pritrjen IR senzor razdalje, ki pove krmiljenju, da je plezalec prijel določen oprimek.

V zadnjo stran so urezani kanali, po katerih so napeljane žice za delovanje krmiljenja. Na desni strani spodaj so tipke za upravljanje deske. Tam se nahaja tudi priključek za napajanje [tip priključka], v katerega se vklopi 12V enosmerne napetosti.

Na sredini se nahaja 20 x 4 LCD prikazovalnik, ki med vajo prikazuje različne parametre. Prikazovalnik je podrobneje opisan v poglavju



slika 26: Oprijemek

Vir: Lasten



slika 27: Najina Plezalna deske

Vir: Lasten

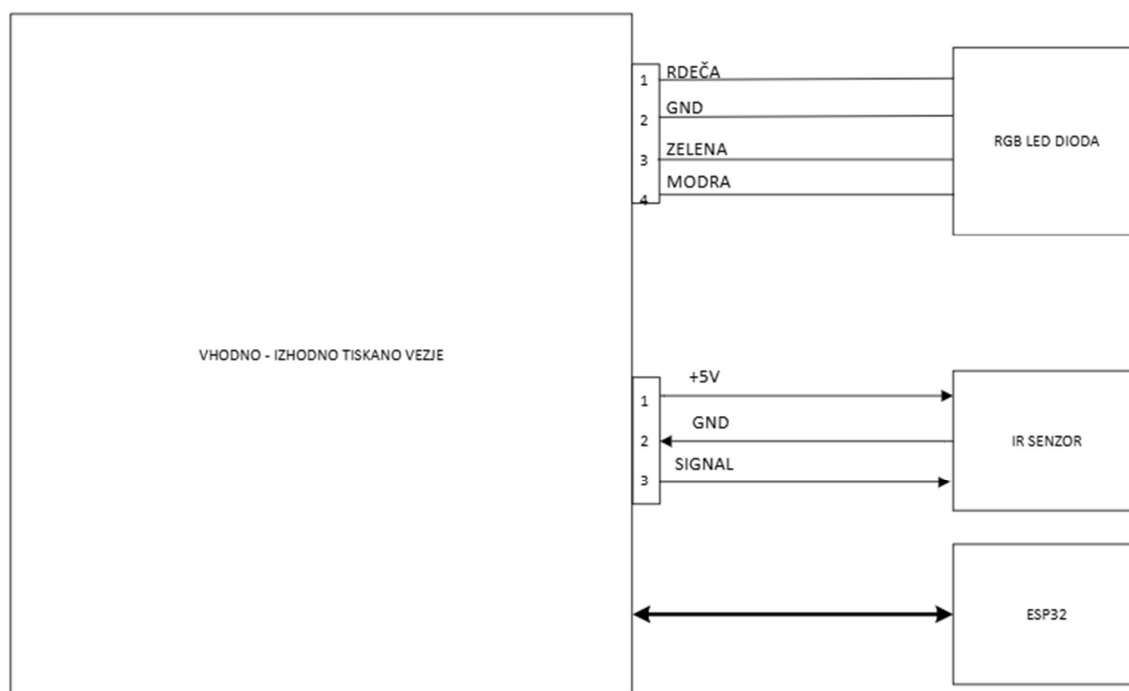
### 3.3 Barvno markiranje oprimkov

Vsak oprimek ima na sredini zgornje stene pritrjeno RGB svetlečo (LED) diodo, ki bo prikazovala status tega oprimka. Te diode bodo imenovane statusne diode.

Statusne diode bodo krmiljene s pomočjo 74HC595 pomičnih registrov. Vsaka barva bo povezana na svoj priključek pomičnega registra. Zaradi nižjega števila zahtevanih komponent ima ena RGB dioda samo en skupni upor povezan med *COMMON* priključkom in maso.

V izogib presežku maksimalnega skupnega toka pomičnega registra bo vsak pomični register krmilil 2 statusni diodi in 2 priključka naslednje.

Povezave si bodo sledile po naslednji logiki:



slika 7: Povezovanje modulov in tiskanih vezji.

Vir: Lasten

PRIKLJUČEK NA 74HC595	PRIKLJUČEK NA DIODI + ŠT. DIODE	BARVA
-----------------------	---------------------------------	-------

Q1	R 1	RDEČA
Q2	G 1	ZELENA
Q3	B 1	MODRA
Q3	R 2	RDEČA
Q4	G 2	ZELENA

74HC595 pomični register ima tudi *ENABLE* priključek, ki spravi izhod na stanje z visoko impedanco. Vsi *ENABLE* priključki vseh pomičnih registrov so povezani skupaj, kar znižuje potrebo število priključkov na mikrokrmilniku.

### 3.4 IR senzor

Krmiljenje zaznava uporabnikove gibe (kateri oprimek prime) s pomočjo IR senzorjev razdalje.

Ta senzor odda logični signal, ko zazna predmet v nastavljenem območju razdalje.



slika 8: IR senzor razdalje

Vir: <https://www.cip.si/ir-senzor-arduino>

Zaradi visokega števila teh senzorjev sva uporabila *input shift register* 74HC165, ki poveča število digitalnih vhodov. IR senzor sva vgradila v oprimek najine plezalne plošče. Senzor služi, da preverja ali je oprimek zaseden, torej ali je roka v njem ali ne.

Senzorji se nahajajo v utorih med dvema deloma posamezne strani deske. Zaradi omejenega prostora so senzorski elementi odstranjeni iz vezja IR senzorja in z vezjem povezani z žic.

IR senzorji preverjajo, ali katerega oprimka se je dotaknil uporabnik. Tako program prepozna, ali uporabnik pravilno izvaja vaje.



slika 25: IR senzor

Vir: Lasten

### 3.5 LCD

Krmilna enota bo z uporabnikom na plošči komunicirala prek LCD zaslona. Kot prikazovalnik je uporabljen 20 x 4 LCD prikazovalnik z I2C krmiljenjem. Vsebina prikazovalnika se bo spreminjala glede na izbrani plezalni program.



Slika 9: 20 x 16 I2C LCD prikazovalnik

Vir:

<https://www.sunfounder.com/cdn/shop/products/sunfounder-default-title-iic-i2c-twi-serial-2004-20x4-lcd-module-shield-for-arduino-uno-mega2560-11137787232417.jpg?v=1768446624>

Prikazovalnik med izvajanjem programa izpisuje naslednje podatke:

- Na vrhu je izpisano ime programa.
- V drugi vrstici levo je prikazana sila na levi strani deske v % celotne sile. Za tem je izpisan zahtevan procent sile na levi strani deske.
- V drugi vrstici desno je prikazana sila na desni strani deske v % celotne sile. Pred tem je izpisan zahtevan procent sile na desni strani deske.
- V tretji vrstici je prikazan čas do konca koraka vaje.
- V četrti vrstici je prikazano, na katerem koraku trenutno smo in koliko korakov je v programu.



slika 10: Meni za izbiranje programov na LCD prikazovalniki  
Vir: Lasten



slika 12: LCD prikazovalnik ob koncu programa  
Vir: Lasten



slika 11: Prikaz parametrov na LCD prikazovalniku med izvajanjem programa. Vir: Lasten

### 3.6 Meritev sile

Za merjenje sile, ki jo plezalec izvaja na posamezno polovico plezalne deske, smo uporabili **tenzometrični merilnik sile (*load cell*)**. Gre za senzor, ki glede na mehansko obremenitev spreminja lastno upornost. Vsak senzor vsebuje 2 tenzometrična merilnika sile, povezana zaporedno. Tako tvorita delilnik napetosti. Nazivna upornost posameznega člena pri obremenitvi 0 kg je  $1000\Omega$ . Napetost se spreminja v za nekaj mV, zato je potrebno ta signal ojačati.



slika 13: Merilnik sile.  
Vir: Lasten

### 3.7 Princip delovanja merilnika sile

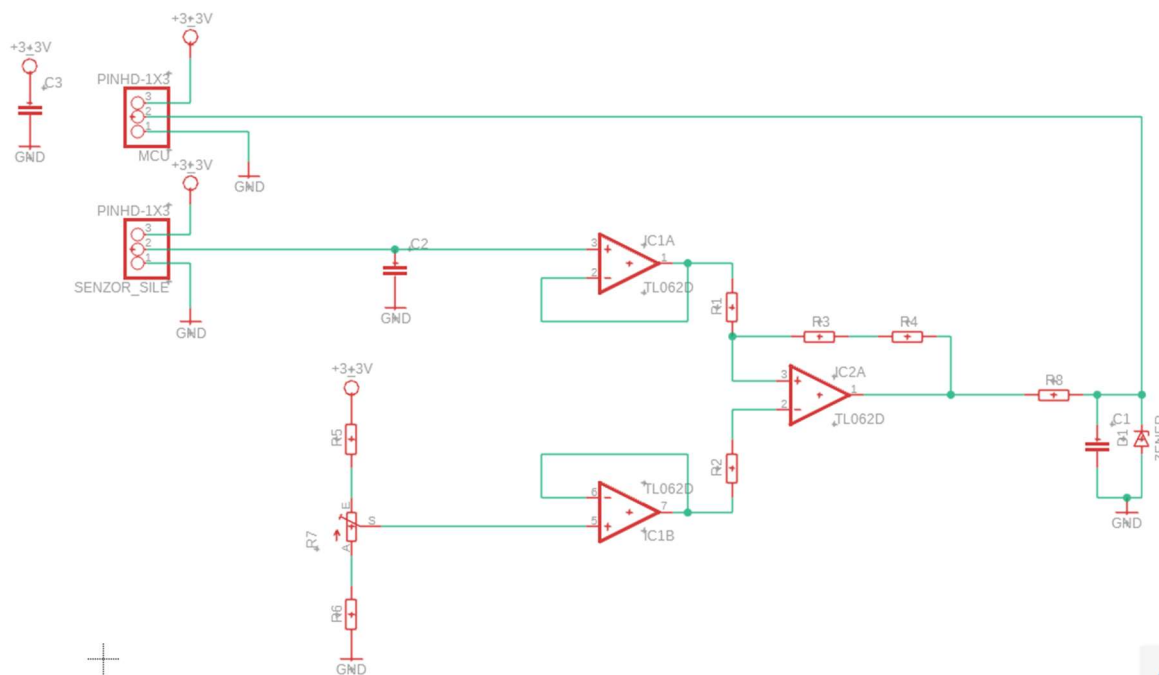
Tenzometrični senzor deluje na principu **Wheatstonovega mostiča**, ki ga sestavljajo štiri tenzometri. Ko na senzor deluje sila, se mehansko telo senzorja rahlo deformira. Ta deformacija povzroči spremembo upornosti tenzometrov, kar vodi do neravnovesja v mostiču in posledično do spremembe izhodne napetosti.

Izhodna napetost senzorja je zelo majhna (v območju milivoltov), zato je potrebna dodatna ojačitev signala.

### 3.8 Ojačitev in obdelava signala

Za ojačenje je uporabljen operacijski ojačevalnik v konfiguraciji diferencialnega ojačevalnika. Pred vhodom diferencialnega ojačevalnika je sledilnik napetosti zaradi lažjega

nastavljanja ojačenja. Za točno nastavitve referenčne točke je na vezju potenciometer RV1, ki spreminja referenčno napetost. Ojačenje je fiksno nastavljeno z upori na pozitivni povratni vezavi. Signal se pretvori v digitalni zapis v integriranem analogno-digitalnem pretvorniku v ESP32.



slika 14: Ojačevalnik za merjenje sile

Vir: Lasten

### 3.9 Umerjanje senzorja

Za pravilno delovanje sistema je bilo potrebno senzor umeriti. Umerjanje smo izvedli z uporabo znanih uteži, pri čemer smo izmerjene vrednosti primerjali z dejanskimi masami. Na podlagi teh meritev smo določili umeritveni faktor, ki omogoča pretvorbo surovih meritev v realne vrednosti sile. Pred tem pa je bilo potrebno nastaviti delovno točko operacijskega ojačevalnika.

### 3.10 Porabnost pri treningu

Porabnost pri treningu predstavlja pomemben vidik pametne plezalne deske, saj neposredno vpliva na zanesljivost delovanja, avtonomijo sistema ter primernost za domačo uporabo. V

okviru raziskovalne naloge smo analizirali porabo električne energije posameznih sklopov sistema med izvajanjem treninga.

Glavni porabnik električne energije v sistemu je mikrokrmilnik ESP32, ki poleg obdelave podatkov senzorjev skrbi tudi za brezžično komunikacijo s strežnikom. Med aktivno Wi-Fi povezavo je poraba toka večja, vendar se ta del delovanja pojavlja predvsem ob zagonu naprave in ob nalaganju trening programov. Med samim izvajanjem vadbe je komunikacija omejena, kar prispeva k zmanjšani porabi energije.

Pomemben del porabe predstavljajo tudi RGB LED diode, ki so aktivne zgolj na oprimkih, vključenih v trenutni korak vadbe. Takšen način krmiljenja omogoča, da niso hkrati vključene vse diode, temveč le tiste, ki so nujno potrebne za vizualno vodenje plezalca. S tem se bistveno zmanjša skupna poraba električne energije in segrevanje sistema.

LCD prikazovalnik porablja relativno malo energije, vendar je stalno vključen med treningom, saj uporabniku prikazuje ključne informacije o izvajanju vaj. Njegov prispevek k skupni porabi je majhen v primerjavi z drugimi komponentami, vendar pomemben z vidika uporabniške izkušnje.

IR senzorji in merilniki sile porabljajo zelo malo energije, saj delujejo z nizkimi tokovi in so večino časa v stanju zaznavanja. Njihova poraba je zanemarljiva v primerjavi z mikrokrmilnikom in LED diodami, vendar imajo ključno vlogo pri zbiranju podatkov o izvedbi treninga.

Analiza porabnosti je pokazala, da je pametna plezalna deska primerna za dolgotrajno uporabo brez tveganja za pregrevanje ali preobremenitev napajalnega sklopa. Sistem je zasnovan tako, da porablja energijo le tam, kjer je to potrebno, kar je posebej pomembno za uporabo v domačem okolju.

Na podlagi ugotovitev lahko zaključimo, da je porabnost pri treningu učinkovito optimizirana in skladna z namenom naprave. V prihodnje bi bilo možno porabo še dodatno zmanjšati z uporabo energijsko varčnih načinov delovanja mikrokrmilnika in z optimizacijo svetilnosti LED diod, vendar že trenutna izvedba zadostuje za varno, zanesljivo in dolgotrajno uporabo pametne plezalne deske.

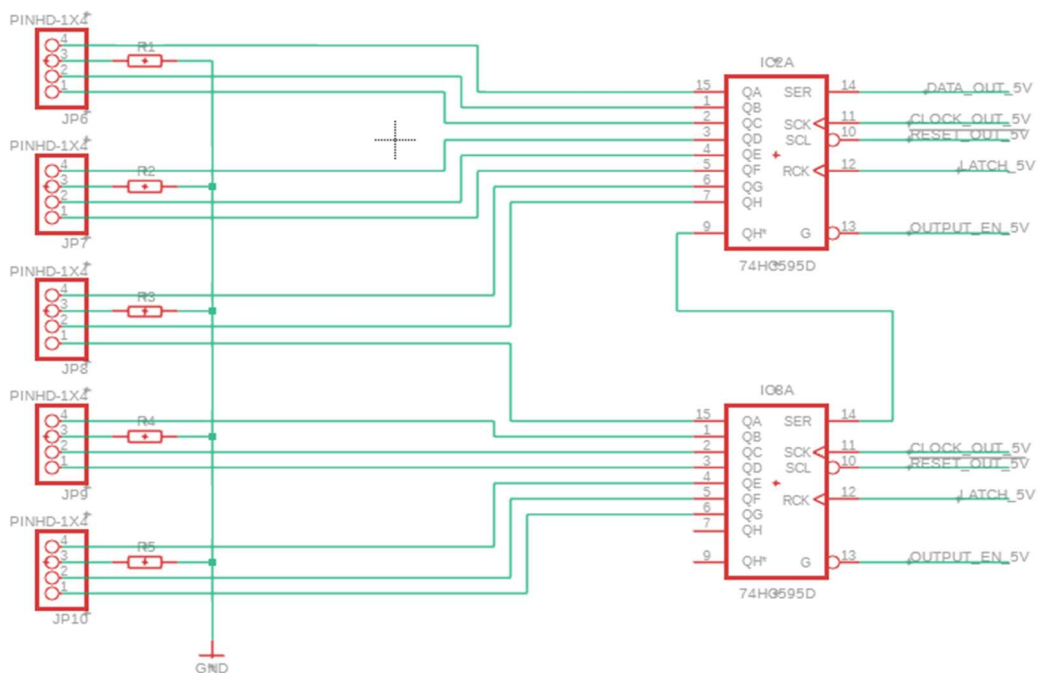
### 3.11 Krmilni sistem

Krmiljenje je zasnovano okoli ESP32 mikrokrmilnika. Ta mikrokrmilnik ima dve procesorski jedri, ki lahko komunicirata med sabo prek skupnih registrov in pomnilnika.

Prvo jedro skrbi za izvajanje programa. Iz pomnilnika bo prebral nastavljen program vaje in ga izvajal v pravilnem časovnem zaporedju.

Drugo jedro skrbi za nalaganje programa vaje, upravljanje plezalne deske z gumbi in komunikacijo z mobilnim telefonom ali računalnikom.

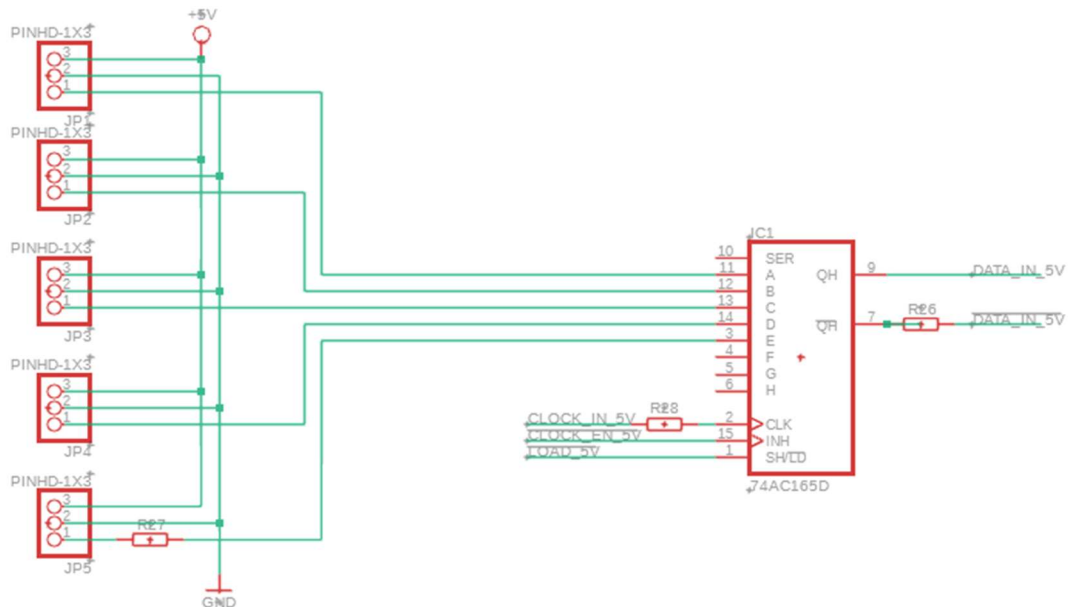
Zaradi potrebe po velikem številu izhodnih priključkov pri krmiljenju RGB LED diod je uporabljen 74HC595 »shift register«, ki pretvori signal iz 3 vhodnih povezav v 8 izhodnih. Njegova prednost je, da jih je mogoče povezati zaporedno. S tem dobim še več izhodnih priključkov z uporabo samo 3 priključkov mikrokrmilnika.



slika 15: Krmiljenje RGB LED diod.

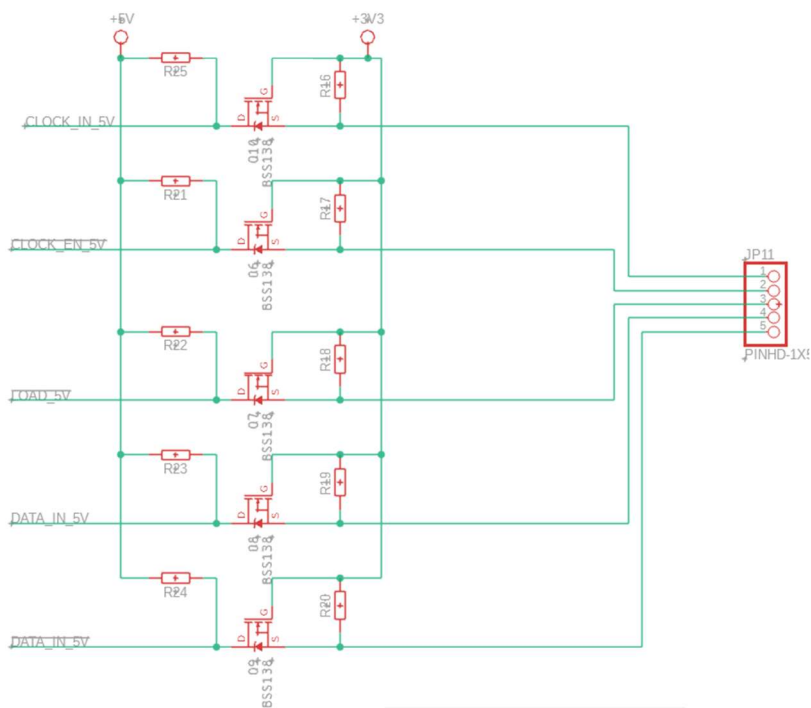
Vir: Lasten

Na enak problem sva naletela pri vhodnih priključkih. Rešitev pa je 74HC165. To je vhodna verzija integriranega vezja 74HC595, ki ima isti princip delovanja.



slika 18: Vhodni pomični register 74HC165 in pripadajoče povezave Vir: Lasten

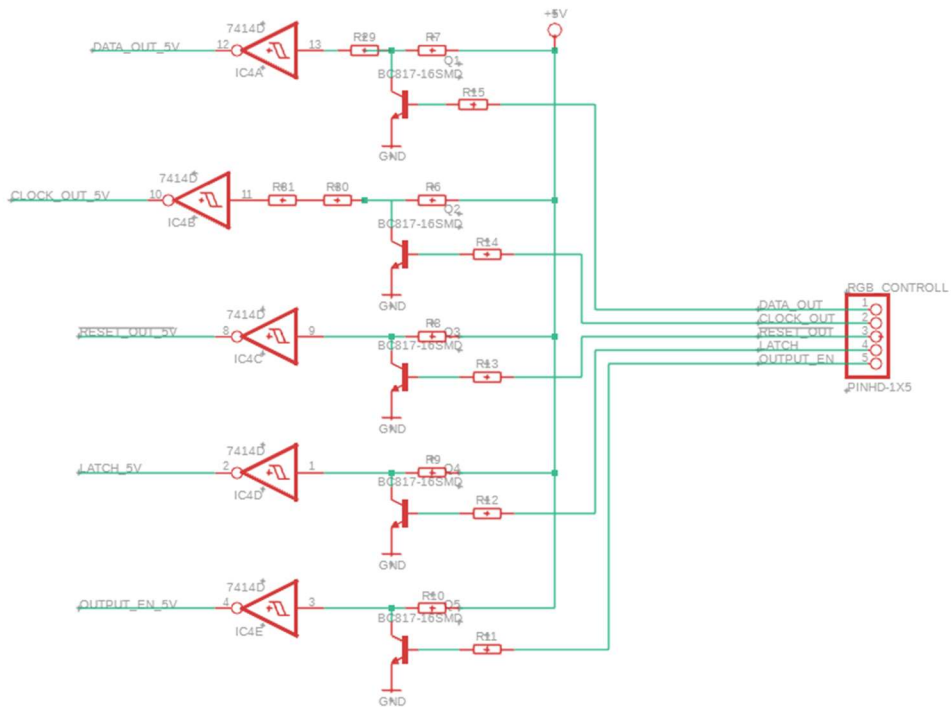
Zgoraj omenjena integrirana vezja delujeta na 5V napajanju in 5V TTL kompatibilnih logičnih vhodih. ESP32 pa deluje na 3,3V. Ta problem sva rešila s pretvornikom napetostnih nivojev. Za pretvornik sva uporabila dva 10kΩ upora in BSS138 N-kanalni MOSFET.



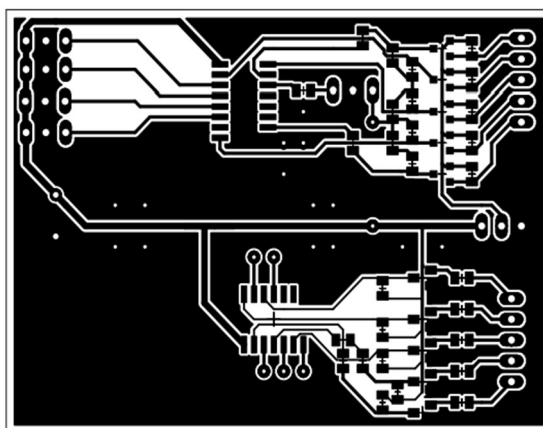
slika 21: Pretvornik s 3,3V na 5V logični signal s BSS138  
Vir: Lasten

Za krmiljenje 74HC165 je uporabljen zgoraj omenjeni MOSFET zaradi tega, ker lahko deluje v obe smeri. Pri 74HC595 ni potrebe po obojestranski komunikaciji, zato sva uporabila BC817-16 tranzistor kot *emitterski* ojačevalnik. Ker to zamakne fazo signala za

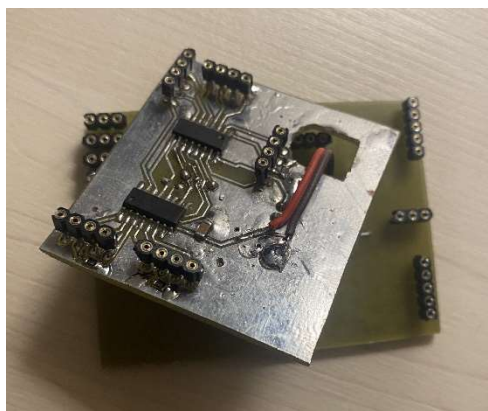
180° je za njim 74HC14 *schmitt trigger* negator, ki zamakne fazo signala za dodatnih 180°, kar pomeni, da bo faza signala na izhodu enaka vhodu.



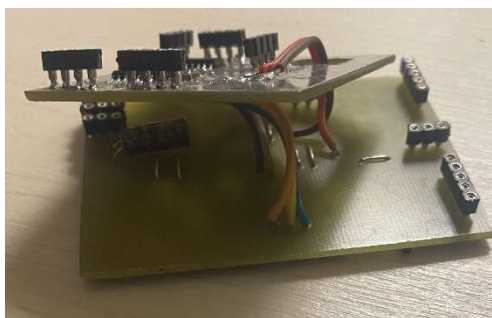
slika 23: Pretvornik z 3,3V na 5V logični signal z BC817-16 in 74HC14  
Vir: Lasten



slika 22: Spodnji del vezja za krmiljenje RGB diod in IR senzorjev  
Vir: Lasten



slika 24: Tiskano vezje za vhode in izhode, zgornja stran.  
Vir: Lasten



slika 25: Povezava zgornjega in spodnjega tiskanega vezja za vhode in izhode.  
Vir: Lasten

Tiskana vezja so narejena doma po postopku termičnega prenosa »tonerja«. Vsa vezja so zaradi lažjega povezovanja enostranska. Načrtovana so bila v programu Autodesk Fusion360, ki vsebuje verzijo EaglePCB.

Vsaka stran plezalne deske vsebuje po 1 komplet takšnih vezji. Vezja sta skrita med dvema deskama.

## 3.12 Program

Plezalna plošča za prikaz programa, uporablja RGB LED diode. Način krmiljenja RGB LED diod za prikaz programa je opisan v poglavju: [3.1.6 Krmilni sistem](#) in [3.1.2. Barvno markiraje oprimkov](#)

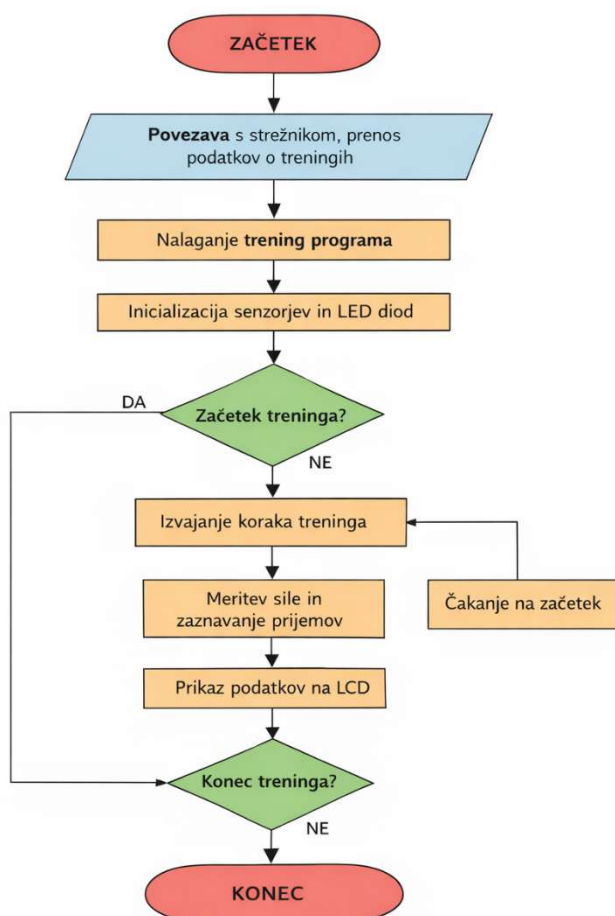
V normalnem stanju so luči za prikaz programa izklopljene v vseh oprimkih. Ko bo plošča prikazala, kateri oprimek mora plezalec prijeto, se oprimek obarva modro. Takrat mora plezalec opraviti neko nalogo (npr.: se držati določen čas, pravilno razporediti silo na roke itd.). Ko plezalec uspešno opravi nalogo, se barva oprimka spremeni v zeleno.

Ob zagonu se ESP32 najprej poveže na strežnik, ki pa mi pošlje .json datoteko, ki jo ESP32 razčleni v posamezne strukture. Takrat ima ESP32 podatke o vseh plezalnih programih.

Takrat se na LCD prikazovalnik izpiše poziv, da naj izberemo plezalni program. Po nekaj sekundah se prikažejo imena vseh plezalnih programov. Dolžina imen je omejena na maksimalno 15 znakov. S tipkama *UP* in *DOWN* se pomikamo po prikazanih programih. S pritiskom na tipko *OK* izberemo program.

Takrat se na prikazovalniku pojavi poziv za začetek plezalnega programa. S ponovnim pritiskom na tipko *OK* začnemo plezalni program. Po nekaj sekundah se bo program začel izvajati.

Med izvajanjem programa se bodo na prikazovalniku izpisovali različni parametri, opisani v poglavju [3.6. LCD](#)



slika 26: Diagram poteka prvega programa. Prva verzija programa je bila napisana po tem diagramu poteka. Vir: Lasten

### 3.13 Mobilna aplikacija

Mobilna aplikacija predstavlja uporabniški vmesnik za upravljanje pametne plezalne deske. Namenjena je tako plezalcem kot trenerjem, saj omogoča nastavitve treningov, izbiro plezalnih programov in spremljanje rezultatov.

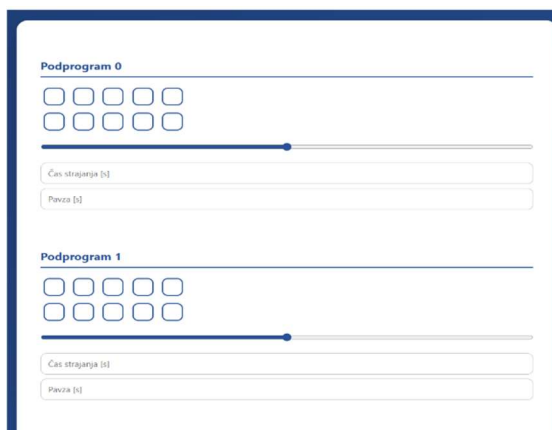
Prek aplikacije lahko trener na daljavo pripravi trening, določi zaporedje oprimkov, trajanje vaj ter zahtevnost. Aplikacija omogoča tudi pregled zgodovine treningov, primerjavo rezultatov in spremljanje napredka posameznega uporabnika.

Mobilna aplikacija deluje v brskalniku na naslovu <https://plez.lovro7.eu>. Ko kliknemo to povezavo, strežnik brskalniku pošlje dokument `add_program.html` (Priloga 12.2.B). Tam sta 2 vpisna polja in tipka »POŠLJI«, ki shrani vpisane podatke. V prvo polje vpišem ime plezalnega programa, v drugo pa število vaj znotraj programa.

Takrat se v zbirki podatkov ustvari nova tabela, s strukturo v prilogi 12.2.6. Vsaka vrstica v tabeli predstavlja nov korak vadbe.

Ko pritisnemo tipko »POŠLJI«, se odpre naslednja spletna stran, v kateri izberemo parametre za posamezni korak. Pojavi se toliko polj za vpis parametrov, kolikor smo jih izbrali na prejšnji strani.

Ko so vsi parametri vpisani, shranimo program. Takrat se na strežniku izvede ukaz »INSERT INTO `IME TABELE`...«. Ta ukaz shrani vse izbrane parametre.



Slika 27: Spletna stran za dodajanje programov.  
Vir: Lasten

### 3.14 Povezljivost

Pametna plezalna deska uporablja brezžično povezavo Wi-Fi, ki jo omogoča mikrokontroler ESP32. Ta povezljivost omogoča komunikacijo med plezalno desko in mobilno aplikacijo, ne glede na to, ali se uporabnik nahaja v istem omrežju ali na daljavo.

S tem je omogočen nadzor trenerja na daljavo, prenos podatkov o treningu ter sprotno posodabljanje nastavitve vadbe brez fizičnega dostopa do naprave.

V teoriji lahko ESP32 pošilja podatke direktno v mobilno napravo, a tam nastane problem. IP naslovi se samodejno posodablajo. To pomeni, da bi recimo ESP32 poslal paket podatkov na mobilno napravo, ki bi ga tudi sprejela, nato pa bi se napravi spremenil IP naslov, zato podatki ne bi bili dostavljeni na mobilno napravo.

To sva rešila z uporabo strežnika, ki ima statični IP naslov. Tako v ESP32 vpišemo povezavo do strežnika in moje domene (<http://plez.lovro7.eu>).

Tako je povezava rešena s preprostim http protokolom. Podatki niso zakodirani, ker ESP32 ne omogoča dekodiranja in http protokola.

ESP32 ob vklopu prek povezave [http://plez.lovro7.eu/get\\_wrkt.php](http://plez.lovro7.eu/get_wrkt.php) pridobi .json datoteko, ki vsebuje vse do sedaj shranjene plezalne programe. Struktura .json datoteke je naslednja:

```
{
  "IME PROGRAMA": [
    {
      "num": ZPOREDNO ŠTEVILO (INDEX) PROGRAMA
      "grip_l": LEVI OPRIMEK
      "grip_r": DESNI OPRIMEK
      "balanse": RAZMERJE LEVE IN DESNE SILE
      "time": ČAS KORAKA
      "delay": PAVZA DO NASLEDNJEGA KLORAKA
    },
    {
```

```
"num": 1
  "grip_l": 2
  "grip_r": 2
  "balanse": 0
  "time": 3
  "delay": 4
}
```

## 4 KONČNI MODEL

Končni model pametne plezalne deske predstavlja funkcionalno celoto, ki združuje mehansko konstrukcijo, elektronske komponente in programsko opremo. Model je bil izdelan z namenom preverjanja zastavljenih hipotez ter praktične uporabnosti sistema v realnem okolju.

Mehanska konstrukcija plezalne deske je izdelana iz lesa, saj ta material omogoča dobro razmerje med trdnostjo, težo in ceno. Deska je razdeljena na levo in desno polovico, ki sta med seboj mehansko povezani, hkrati pa vsaka stran deluje kot samostojna merilna enota. Takšna zasnova omogoča ločeno merjenje sile, ki jo plezalec izvaja z levo in desno roko, kar je ena izmed ključnih funkcionalnosti sistema. Deska je dimenzionirana tako, da je primerna za montažo v stanovanjske prostore, kar potrjuje začetno zastavljeni cilj majhnosti in praktičnosti.

Na sprednji strani deske so nameščeni oprijemi z integriranimi RGB LED diodami. Te diode služijo kot vizualni prikaz zaporedja prijemov in stanja posameznega koraka vadbe. Barvno vodenje omogoča intuitivno uporabo tudi manj izkušenim plezalcem, saj jasno nakazuje, kateri oprimek je aktiven in ali je bila naloga pravilno izvedena. V oprijemih so dodatno vgrajeni IR senzorji, ki zaznavajo prisotnost roke in s tem omogočajo preverjanje pravilnosti izvedbe vaj.

V osrednjem delu plezalne deske je nameščen LCD prikazovalnik, ki uporabniku med izvajanjem treninga prikazuje ključne informacije, kot so trenutni program, razmerje sil med levo in desno stranjo, čas posameznega koraka ter napredek v programu. Prikazovalnik omogoča neposredno povratno informacijo brez potrebe po uporabi mobilne naprave, kar povečuje uporabnost sistema.

Krmilni del sistema temelji na mikrokrmilniku ESP32, ki skrbi za obdelavo podatkov senzorjev, krmiljenje LED diod, komunikacijo z LCD prikazovalnikom ter brezžično povezavo s strežnikom. Zaradi omejenega števila vhodno-izhodnih priključkov so uporabljeni pomični registri, ki omogočajo učinkovito razširitev sistema brez povečanja kompleksnosti vezja. Elektronski del je zasnovan modularno, kar omogoča lažje vzdrževanje in morebitne nadgradnje.

Programsko je sistem razdeljen na več funkcionalnih sklopov. Eden skrbi za izvajanje plezalnih programov in časovno usklajevanje korakov, drugi pa za komunikacijo z uporabnikom in strežnikom. Treningi so shranjeni na strežniku v obliki strukturiranih podatkov, ki se ob zagonu naprave prenesejo na plezalno desko. Takšen način delovanja omogoča enostavno posodabljanje in prilagajanje treningov brez fizičnega posega v napravo.

Končni model pametne plezalne deske izpolnjuje vse osnovne zahteve, ki smo si jih zastavili na začetku raziskovalne naloge. Sistem je funkcionalen, zanesljiv in cenovno dostopen, hkrati pa omogoča dovolj prilagodljivosti za nadaljnji razvoj. Model dokazuje, da je mogoče z relativno preprosto in cenovno ugodno tehnologijo izdelati pametno športno napravo, ki uporabniku nudi napredne funkcionalnosti in kakovostno podporo pri treningu.

## 5 REZULTATI

Rezultati raziskovalne naloge so pokazali, da je razvita pametna plezalna deska uspešno dosegla zastavljene cilje in potrdila večino začetnih hipotez. Končni model omogoča funkcionalno, zanesljivo in ponovljivo izvajanje treningov, pri čemer uporabniku zagotavlja jasne povratne informacije o izvedbi vaj.

Med testiranjem je sistem pravilno zaznaval prijeme oprimkov s pomočjo IR senzorjev in natančno meril silo na levi ter desni polovici plezalne deske. Meritve so se izkazale za dovolj stabilne, da je bilo mogoče spremljati razporeditev obremenitve med rokama in zaznati morebitna neravnovesja pri dvigovanju. Prikaz podatkov na LCD zaslonu je omogočal sprotno spremljanje ključnih parametrov, kot so čas posameznega koraka, razmerje sil in napredek v programu.

RGB LED diode so se izkazale kot zelo učinkovit način vizualnega vodenja plezalca. Plezalci so lahko hitro in intuitivno prepoznali, kateri oprimek je aktiven in kakšen je njegov status. Barvno kodiranje (modra za aktivni oprimek, zelena za uspešno opravljen korak) je bistveno prispevalo k preglednosti in razumljivosti treninga.

Pomemben rezultat je tudi uspešno delovanje brezžične povezave med plezalno desko in strežnikom. ESP32 je ob zagonu zanesljivo pridobil podatke o treningih v obliki JSON datotek, kar je omogočilo nalaganje in izvajanje različnih plezalnih programov brez potrebe po ponovnem programiranju naprave. To potrjuje, da sistem podpira koncept treninga na daljavo.

Na podlagi testiranj lahko ugotovimo, da pametna plezalna plošča omogoča zbiranje uporabnih podatkov, ki so primerni za analizo napredka plezalca, prilagajanje treningov in dolgoročno spremljanje razvoja moči ter tehnike.

## 6 FINANČNO VREDNOTENJE MODELA

Pri finančnem vrednotenju modela smo analizirali stroške materiala, elektronskih komponent, mehanske izdelave ter sestava prototipa pametne plezalne deske. Poseben poudarek smo namenili primerjavi s komercialno dostopnimi pametnimi plezalnimi sistemi, ki so trenutno na trgu.

Stroški izdelave prototipa vključujejo leseno plezalno desko, merilnike sile, IR senzorje, RGB LED diode, mikrokontrolnik ESP32, pomične registre, LCD prikazovalnik, napajalni sklop ter osnovne elektronske komponente (upori, tranzistorji, konektorji in tiskano vezje). Ker je bila večina komponent kupljena posamično in ne v industrijskih količinah, so ocenjeni stroški nekoliko višji, kot bi bili pri serijski proizvodnji.

Kljub temu skupna cena izdelave prototipa ostaja bistveno nižja od primerljivih komercialnih rešitev, katerih cene se gibljejo od nekaj sto do več tisoč evrov. Najina rešitev združuje več naprednih funkcij, kot so merjenje sile na vsaki roki, zaznavanje prijemov, vizualno vodenje z LED diodami in povezljivost z aplikacijo, ob tem pa ostaja cenovno dostopna.

Pomemben vidik finančnega vrednotenja je tudi razširljivost sistema. Programska zasnova omogoča nadgradnjo brez dodatnih večjih stroškov strojne opreme, kar pomeni, da se lahko funkcionalnosti izboljšujejo predvsem s programsko opremo. Prav tako bi se v primeru serijske izdelave stroški posamezne enote bistveno znižali.

Na podlagi opravljene analize lahko zaključimo, da pametna plezalna deska izpolnjuje cilj cenovne dostopnosti. Primerna je za manjše plezalne klube, športna društva ter posameznike, ki si želijo naprednega treninga, vendar si ne morejo privoščiti dragih komercialnih sistemov. Finančno vrednotenje tako potrjuje, da je bila izbira materialov in tehničnih rešitev ustrezna in skladna z zastavljenimi cilji raziskovalne naloge.

## 7 PREIZKUS S PLEZALCI

Preizkus pametne plezalne deske je bil izveden s pomočjo plezalcev različnih nivojev znanja. Plezalci so testirali različne trening programe in podali povratne informacije o uporabnosti, zahtevnosti in ergonomiji naprave.

Rezultati preskusa so pokazali, da je sistem intuitiven za uporabo in primeren tako za začetnike kot tudi za bolj izkušene plezalce.

Navodila trenerja Gorazda Hrena

Gorazd Hren je trener športnega plezanja iz Velenja, ki s svojim strokovnim znanjem in izkušnjami pomembno prispeva k razvoju plezalnih sposobnosti mladih ter rekreativnih športnikov. Pri svojem delu poudarja varnost, pravilno tehniko in postopno napredovanje. S svojo pripravljenostjo pomagati in deliti znanje je nudil dragoceno podporo tudi pri pripravi raziskovalne naloge. Privolil je v intervju, kjer sva mu predstavila najino idejo, nad katero je bil zelo navdušen. Svetoval nama je s premerom lukenj in priporočal je, da nadrino debelino oprimkov 1mm, 1.3 mm, 1.5 mm, in 2 mm. Zelo mu je bila všeč najina ideja, da bo plošča merila, s kakšno silo se plezalec dvigne in ali se na desni in levi roki dvigne enakomerno. Podal nama je nekaj idej, kaj vse bi lahko še dodala plezalni deski. Edina stvar, ki mu na najini plezalni deski ni bila preveč všeč, je funkcija, da bi se moral plezalec prijemat na tiste oprimke, kjer se zasveti lučka, rekel je, da je sicer zelo dobra ideja, ampak da profesionalni plezalci tega ne bi uporabljali. Bi pa to bil nek zabaven način, kako lahko svoj trening popestri rekreativni plezalec.

## 8 RAZPRAVA

V razpravi smo analizirali dosežene rezultate in jih primerjali z začetnimi hipotezami. Večina hipotez se je izkazala za pravilne, nekatere rešitve (npr. uporaba ultrazvočnega senzorja) pa zahtevajo dodatno optimizacijo.

V razpravi smo analizirali dosežene rezultate in jih primerjali z začetnimi hipotezami. Večina hipotez se je izkazala za pravilne, nekatere rešitve (npr. uporaba ultrazvočnega senzorja) pa zahtevajo dodatno optimizacijo.

Za prvo hipotezo, ki je bila, da je najina plošča primerna za montažo v stanovanju, sva jo dala najinemu sošolcu, ki si jo je namestil v sobo in jo še danes uporablja. Rekel je, da je montaža bila zelo enostavna in da ni bojazni, da bi padla s stene. Tako da je hipoteza potrjena.

Druga hipoteza: sošolec, ki je testiral ploščo, je treniral z njo med treningom. Opazovala sva njegov trening. Na aplikaciji se je prikazovalo vse, kar je počel, torej oprimki, katere je prijemal, sila, s katero se je dvigoval in tudi čas, kako hitro se je dvignil. Tako da lahko tudi to hipotezo potrdiva.

Tretja hipoteza: taisti sošolec je imel plezalno desko na testiranju približno en mesec. Med testiranjem je dejal, da je merjenje sile leve in desne roke zelo dobra funkcija. Dejal je, da se mu je med uporabo deske zelo izboljšalo enakomerno dvigovanje. Za sledenje oprimkov po nekem zaporedju je dejal, da je zelo zanimiva funkcija, ki popestri trening. Funkcija merjenje časa mu je bila zelo všeč, ker si je s tem zelo izboljšal čas dviga. Tako, da lahko hipotezo potrdiva.

Četrta hipoteza: ko nama je vrnil desko, sva ga vprašala, če je deska omogočala učinkovito treniranje in je rekel, da je z najino pametno plezalno desko lahko opravil učinkovit trening. To pomeni, da lahko tudi to hipotezo potrdiva.

## 9 ZAKLJUČEK

V okviru raziskovalne naloge smo razvili in preizkusili pametno plezalno desko, katere glavni namen je omogočiti kakovosten in učinkovit trening plezanja v manjših prostorih, z možnostjo nadzora na daljavo. Izhajali smo iz dejstva, da so obstoječe pametne plezalne rešitve na trgu pogosto zelo drage, prostorsko zahtevne in zato težko dostopne manjšim plezalnim klubom ter posameznikom. Naš cilj je bil razviti cenovno dostopnejšo in funkcionalno alternativno rešitev.

Raziskava je pokazala, da je s premišljeno izbiro materialov, elektronskih komponent in programske opreme mogoče izdelati pametno plezalno napravo, ki združuje več naprednih funkcij. Med najpomembnejše sodijo vizualno vodenje plezalca z RGB LED diodami, merjenje sile na posamezni roki, zaznavanje prijmov ter brezžična povezljivost z mobilno aplikacijo. Takšen sistem omogoča tako samostojno vadbo kot tudi sodelovanje s trenerjem na daljavo.

Med razvojem smo uspešno preverili večino zastavljenih hipotez. Ugotovili smo, da je plezalna deska zaradi svojih dimenzij primerna za montažo v stanovanjske prostore, da omogoča nastavitve in spremljanje treningov na daljavo ter da lahko plezalcu nudi uporabne povratne informacije o njegovi izvedbi. Posebej pomembna se je izkazala možnost ločenega merjenja sile leve in desne roke, saj ta funkcija omogoča zaznavanje neravnovesij in izboljšanje tehnike plezanja.

Praktični preskusi s plezalcami so pokazali, da je sistem intuitiven za uporabo in primeren za različne ravni znanja. Uporabniki so pozitivno ocenili jasnost vizualnih signalov, uporabnost zbranih podatkov ter možnost prilagajanja treningov. Finančna analiza je potrdila, da je strošek izdelave prototipa bistveno nižji od primerljivih komercialnih rešitev, kar dodatno potrjuje smiselnost zastavljenega cilja.

Kljub uspešnim rezultatom pa projekt odpira tudi možnost za nadaljnje izboljšave. V prihodnje bi bilo smiselno izboljšati mobilno aplikacijo, razširiti nabor trening programov, izboljšati analizo zbranih podatkov ter dodatno optimizirati mehansko zasnovo deske. Prav tako bi bilo zanimivo raziskati uporabo dodatnih senzorjev za še natančnejše spremljanje gibanja plezalca.

Na koncu lahko zaključimo, da pametna plezalna deska predstavlja uspešen primer povezave elektrotehnike, programiranja in športa. Projekt dokazuje, da je mogoče s sodobno tehnologijo izboljšati športni trening in ga narediti dostopnejšega širšemu krogu uporabnikov. Raziskovalna naloga je izpolnila zastavljene cilje in predstavlja dobro osnovo za nadaljnji razvoj ter morebitno praktično uporabo v plezalnih okoljih.

## 10 VIRI IN LITERATURA

V raziskovalni nalogi sva uporabila naslednje vire in literaturo:

- <https://github.com/bblanchon/ArduinoJson>
- <https://github.com/amcewen/HttpClient>

## **11 ZAHVALA**

Zahvaljujema se mentorju Petru Vrčkovniku, ki naju je uvedel v raziskovalno delo in nama s svojimi izkušnjami in znanjem pomagal pri nastajanju raziskovalne naloge.

Zahvaljujema se tudi Bojani Vrbnjak za lektoriranje.

Zahvaljujema se tudi gospe Simoni Diklič, za prevod povzetka v angleščino.

Zahvaljujema se pa še gospodu Gorazdu Hrenu, ki nama je pomagal in svetoval z najino raziskovalno nalogo.

## 12 PRILOGE

### 12.2 Program za esp34

#### 12.1.A. Priloga A - main.cpp

```
#include <Arduino.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <LiquidCrystal_I2C.h>

const char* url = "http://plez.lovro7.eu/get_wrkt.php";

// ===== NASTAVITVE =====

#define DEBUG //Definicija za izpis podatkov na serijski monitor

#define GRIP_N 5 //Stevilo oprimkov na vsaki strani
#define MAX_PROGRAMS 10 //Najvisje stevilo programov
#define MAX_STEPS 20 //Najvisje stevilo korakov v enem programu
#define MAX_NAME_LEN 15 //Najvisja dolzina imena programa
#define MAX_WIFI_CONNECT_TIME 10 //Najvisji cas vzpostavljanja WiFi povezave

//Barve
#define NONE 0
#define RED 1
#define GREEN 2
#define BLUE 3

//Izhoi IR senzorja
#define GRIP_PRESSED 0
#define GRIP_DEPRESSED 1

#define MWCT (MAX_WIFI_CONNECT_TIME * 2) //Casovna konstanta povezovanja WiFi

// ===== PINI =====
#define OK_BTN 4
```

```
#define UP_BTN 16
#define DWN_BTN 17
#define LEFT_SER_OUT 25
#define LEFT_DAT_OUT 26
#define LEFT_LATCH_OUT 27

#define RIGHT_SER_OUT 0
#define RIGHT_DAT_OUT 2
#define RIGHT_LATCH_OUT 15

#define LEFT_LATCH_IN 18
#define LEFT_CLOCK_IN 19
#define LEFT_DATA_IN 20

#define RIGHT_LATCH_IN 21
#define RIGHT_CLOCK_IN 22
#define RIGHT_DATA_IN 23

// ===== STRUKTURE =====

void setup();

//Struktura za korak plezalnega programa
struct WorkoutStep {
    int num;
    int grip_l;
    int grip_r;
    int balanse;
    int time;
    int delay;
};

//Struktura plezalnega programa
struct Workout {
    char name[MAX_NAME_LEN];
    WorkoutStep steps[MAX_STEPS];
    size_t stepCount;
```

```
};

//List plezalnih programov
Workout programs[MAX_PROGRAMS];

//Ostale konstante in spremenljivke
size_t programCount = 0;
int selected_program = 255;
bool left_grip_stat[GRIP_N] = {0, 0, 0, 0, 0};
bool right_grip_stat[GRIP_N] = {0, 0, 0, 0, 0};
const uint8_t colors[4] = {0b000, 0b001, 0b010, 0b100};
uint8_t left_colors[GRIP_N] = {NONE, NONE, NONE, NONE, NONE};
uint8_t right_colors[GRIP_N] = {NONE, NONE, NONE, NONE, NONE};

LiquidCrystal_I2C lcd(0x27, 20, 4);

// ===== FUNKCIJA ZA PRENOS =====

bool downloadAndParseJSON() {
    //Ustvari HTTP stranko in zacne HTTP sejo
    HTTPClient http;
    http.begin(url);

    int httpCode = http.GET(); //Pridobi HTTP kodo
    if (httpCode != HTTP_CODE_OK) {
        #ifdef DEBUG
            Serial.print("HTTP napaka: ");
            Serial.println(httpCode);
        #endif
        http.end(); //Zakljuci HTTP sejo
        return false;
    }

    String payload = http.getString(); //Pridobi odgovor spletne strani
    http.end(); //Zakljuci HTTP sejo
    #ifdef DEBUG
        Serial.println("Prejet JSON:");
    #endif
}
```

```
        Serial.println(payload);
    #endif

    DynamicJsonDocument doc(8192); //Ustvari JSON dokument v knjiznici

    DeserializationError error = deserializeJson(doc, payload); //Preveri JSON za napake
    if (error) {
        #ifdef DEBUG
            Serial.print("JSON napaka: ");
            Serial.println(error.c_str()); //Izpiše napako
        #endif
        return false;
    }

    JsonObject root = doc.as<JsonObject>(); //Ustvari mapo datoteke
    programCount = 0; //Resetira stevec programov

    for (JsonPair kv : root) {

        if (programCount >= MAX_PROGRAMS) break; //Zaključni, ko so obdelani vsi programi

        //Shrani ime programa
        strncpy(programs[programCount].name,
                kv.key().c_str(),
                MAX_NAME_LEN - 1);

        programs[programCount].name[MAX_NAME_LEN - 1] = '\0'; //Nastavi končni znak
        (0x00)
        programs[programCount].stepCount = 0; //Resetira stevec korakov znotraj programa

        JsonArray stepsArray = kv.value().as<JsonArray>(); //Pridobi korake znotraj
        programa

        for (JsonObject step : stepsArray) {

            if (programs[programCount].stepCount >= MAX_STEPS) break; //Zaključni, ce so
            obdelani vsi koraki
```

```
//Vpise vsebino korakov
WorkoutStep &s =
    programs[programCount].steps[programs[programCount].stepCount];

s.num      = step["num"]      | 0;
s.grip_l   = step["grip_l"]   | 0;
s.grip_r   = step["grip_r"]   | 0;
s.balanse  = step["balanse"]  | 0;
s.time     = step["time"]     | 0;
s.delay    = step["delay"]    | 0;

    programs[programCount].stepCount++; //Poveca stevec korakov znotraj programa
}

programCount++; //Poveca stevec programov
}

return true; //Vrne true (0x01). To pomeni, da so podatki točni
}

void displayMarker(int pos, int prevPos){
    //Narise nov marker na novi poziciji
    if(prevPos >= 0 && prevPos < 4){
        lcd.setCursor(18, prevPos);
        lcd.print(" ");
    }
    //Izbrise stari marker
    if(pos >= 0 && pos < 4){
        lcd.setCursor(18, pos);
        lcd.print("<-");
    }
}

void displayPrograms(int offset, int mark){
    int lines = (programCount - offset > 4) ? 4 : (programCount - offset); //Izracuna
    stevilo linje
    //Izpis linje
```

```
    for(int i = 0; i < lines; i++){
        lcd.setCursor(0, i);
        lcd.print("                ");
        lcd.setCursor(0, i);
        lcd.print(programs[i + offset].name);
    }
}

int selectProgram(){
    //Nastavi lokalne spremenljivke
    int index = 0;
    int prevMarker = -1;
    int offset = 0;

    //Izpiše poziv na prikazovalnik
    lcd.clear();
    lcd.print("Izberite program!");
    delay(2000);
    lcd.clear();

    //Izpiše imena programov
    displayPrograms(offset, index);
    displayMarker(index - offset, prevMarker);
    prevMarker = index - offset;

    //Zanka za izbiranje programov
    while(digitalRead(OK_BTN)){
        //Bere stanje tipk in prilagaja prikaz na prikazovalniku
        if(!digitalRead(UP_BTN) && index > 0){
            index--;
            if(index < offset) offset = index;
            displayPrograms(offset, index);
            displayMarker(index - offset, prevMarker);
            prevMarker = index - offset;
            while(!digitalRead(UP_BTN));
        }
        if(!digitalRead(DWN_BTN) && index < programCount - 1){
```

```

        index++;
        if(index >= offset + 4) offset = index - 3;
        displayPrograms(offset, index);
        displayMarker(index - offset, prevMarker);
        prevMarker = index - offset;
        while(!digitalRead(DWN_BTN));
    }
}
while(!digitalRead(OK_BTN)); //Pocakam da izpustimo tipko
return index;
}

void colorLeftGrip(){
    //                MSB                LSB
    uint32_t dat = 0; // 0 0 0...0 C5 C4 C3 C2 C1
    //Pretvori zapise barv v podatke za 74HC595
    for(int i = 0; i < GRIP_N; i++){
        dat |= (left_colors[i] << (i * 3));
    }
    for(int i = 0; i < GRIP_N; i++){
        dat &= ~(left_colors[i] << (i * 3));
    }
    //Izpiše podatke na izhodih 74HC595
    digitalWrite(LEFT_LATCH_OUT, 1);
    shiftOut(LEFT_DAT_OUT, LEFT_SER_OUT, MSBFIRST, (uint8_t)((dat >> 8) & 0xFF));
    shiftOut(LEFT_DAT_OUT, LEFT_SER_OUT, MSBFIRST, (uint8_t)(dat & 0xFF));
    digitalWrite(LEFT_LATCH_OUT, 0);
}

void colorRightGrip(){
    uint32_t dat = 0;
    //Pretvori zapise barv v podatke za 74HC595
    for(int i = 0; i < GRIP_N; i++){
        dat |= (right_colors[i] << (i * 3));
    }
    for(int i = 0; i < GRIP_N; i++){
        dat &= ~(right_colors[i] << (i * 3));
    }
}

```

```
    }  
    //Izpiše podatke na izhodih 74HC595  
    digitalWrite(RIGHT_LATCH_OUT, 1); //Pripravi 74H595 za vpis novih vrednosti  
    shiftOut(RIGHT_DAT_OUT, RIGHT_SER_OUT, MSBFIRST, (uint8_t)((dat >> 8) & 0xFF)); //Iz  
vrednosti vzame drugih 8 bitov in jih pošlje v 74HC595  
    shiftOut(RIGHT_DAT_OUT, RIGHT_SER_OUT, MSBFIRST, (uint8_t)(dat & 0xFF)); //Iz  
vrednosti vzame drugih 8 bitov in pošlje v 74HC595  
    digitalWrite(RIGHT_LATCH_OUT, 0); //Prenese vrednosti na izhod 74HC595  
}  
  
void getLeftGripStat(){  
    //Zajame stanja vhodnih priključkov  
    digitalWrite(LEFT_LATCH_IN, 0);  
    delayMicroseconds(5);  
    digitalWrite(LEFT_LATCH_IN, 1);  
    delayMicroseconds(10);  
    //Prebere vrednosti  
    uint8_t dat = shiftIn(LEFT_DATA_IN, LEFT_CLOCK_IN, MSBFIRST);  
    //Vsak bit pridobljene vrednosti zapise v seznam  
    for(int i = 0; i < GRIP_N; i++){  
        left_grip_stat[i] = (bool)((dat >> i) & 0x01);  
    }  
}  
  
void getRightGripStat(){  
    //Zajame stanja vhodnih priključkov  
    digitalWrite(RIGHT_LATCH_IN, 0);  
    delayMicroseconds(5);  
    digitalWrite(RIGHT_LATCH_IN, 0);  
    delayMicroseconds(10);  
    //Prebere vrednosti  
    uint8_t dat = shiftIn(RIGHT_DATA_IN, RIGHT_CLOCK_IN, MSBFIRST);  
    //Vsak bit pridobljene vrednosti zapise v seznam  
    for(int i = 0; i < GRIP_N; i++){  
        right_grip_stat[i] = (bool)((dat >> i) & 0x01);  
    }  
}
```

```
void updateLCD(int left_force, int right_force, int time_left, int step_n){
    //Na prikazovalnik izpise podatke o trenutnem koraku in programu
    lcd.setCursor(4, 0);
    lcd.print(programs[selected_program].name);
    lcd.setCursor(0, 1);
    lcd.print(left_force);
    lcd.print("% ");
    lcd.setCursor(16, 1);
    lcd.print(right_force);
    lcd.print("% ");
    lcd.setCursor(9, 2);
    lcd.print(time_left);
    lcd.print(" ");
    lcd.setCursor(8, 3);
    lcd.print(step_n + 1);
    lcd.print("/");
    lcd.print(programs[selected_program].stepCount);
    lcd.print(" ");
}

bool checkOtherTrue(bool arr[GRIP_N], int index) {
    for (int i = 0; i < GRIP_N; i++) {
        if (i != index && arr[i]) {
            return 1;
        }
    }
    return 0;
}

void excecuteStep(WorkoutStep s, int step_n){
    unsigned long start_time = millis(); //Shrani cas zacetka vaje v [ms]
    unsigned long prev_time = 0; //Inializira lokalno spremenljivko
    int left_f = 100 - s.balance; //Izrauna silo na levi strani
    int right_f = s.balance; //Izracuna silo na desni strani
    lcd.clear(); //Pocisti prikazovalnik
    left_colors[s.grip_l] = BLUE; //Nastavi barvo pravega oprimka
    right_colors[s.grip_r] = BLUE; //Nastavi barvo pravega oprimka
```

```
colorLeftGrip(); //Nastavi barvo na izhodu
colorRightGrip(); //Nastavi barvo na izhodu
//Zanka, dogler traja korak
while ((millis() - start_time) < s.time * 1000){
    //Vsako sekundo posodobi podatke na zaslonu
    if((millis() - prev_time) > 1000){
        prev_time = millis();
        updateLCD(left_f, right_f, (s.time - ((millis() - start_time) / 1000.)),
step_n);
    }
    getLeftGripStat(); //Pridobi podatke o senzorjih na levi strani
    getRightGripStat(); //Pridobi podatke o senzorjih na desni strani
    //Nastavi barvo oprimkov
    if(checkOtherTrue(left_grip_stat, s.grip_l)){
        left_colors[0] = RED;
        left_colors[1] = RED;
        left_colors[2] = RED;
        left_colors[3] = RED;
        left_colors[4] = RED;
        left_colors[s.grip_l] = BLUE;
        getLeftGripStat();
        getRightGripStat();
    }
    if(checkOtherTrue(right_grip_stat, s.grip_r)){
        right_colors[0] = RED;
        right_colors[1] = RED;
        right_colors[2] = RED;
        right_colors[3] = RED;
        right_colors[4] = RED;
        right_colors[s.grip_r] = BLUE;
        getLeftGripStat();
        getRightGripStat();
    }
}
}
```

```
void measureBalance(){
    lcd.clear(); //Pocisti prikazovalnik
    //Nastavi položaj in izpise imena podatkov
    lcd.setCursor(2, 0);
    lcd.print("Meritev balansa");
    lcd.setCursor(5, 1);
    lcd.print("Izhod: <OK>");
    //Dokler ne pritisnemo tipke <OK>
    while(digitalRead(OK_BTN)){
        //Izpisuje podatke
        lcd.setCursor(0, 2);
        lcd.print(left_force);
        lcd.print("kg ");
        lcd.setCursor(16, 2);
        lcd.print(right_force);
        lcd.print("kg ");
        lcd.setCursor(2, 2);
        lcd.print("Teza: ");
        lcd.print(left_force + right_force);
        lcd.print("kg ");
    }
}

void getPrograms(){
    //Ce uspesno prodpbi in razcleni json datoteko izpise podatke o programu in korakih
    if (downloadAndParseJSON()) {
        #ifdef DEBUG
            Serial.println("\n=== Izpis vseh programov ===");
        #endif
        for (size_t p = 0; p < programCount; p++) {
            #ifdef DEBUG
                Serial.print("\nProgram: ");
                Serial.println(programs[p].name);
            #endif
            for (size_t i = 0; i < programs[p].stepCount; i++) {
                WorkoutStep &s = programs[p].steps[i];
                #ifdef DEBUG
```

```
        Serial.printf(
            " Korak %d | L=%d R=%d B=%d T=%d D=%d\n",
            s.num,
            s.grip_l,
            s.grip_r,
            s.balanse,
            s.time,
            s.delay
        );
    #endif
}
}
else {
    //Izpis morebitne napake
    lcd.clear();
    #ifdef DEBUG
        Serial.println("Napaka pri branju programa!");
    #endif
    lcd.setCursor(1, 0);
    lcd.print("Napaka pri branju");
    lcd.setCursor(1, 1);
    lcd.print("podatkov");
    lcd.setCursor(1, 2);
    lcd.print("<UP> + <DOWN> Osvezi");
}
}

void selectPrpgramToRun(){
    selected_program = selectProgram(); //Pridobi zaporedno številjko (index) izbranega programa
    //Pripravi LCD prikazovalnik
    lcd.clear();
    lcd.home();
    lcd.print("Izbrali ste program:");
    lcd.print(programs[selected_program].name); //Izpiše ime izbranega programa
    lcd.print("Pritisni <OK>");
}
```

```
while(digitalRead(OK_BTN)); //Pocaka na priitisk tipke <OK>
size_t stepCount = programs[selected_program].stepCount;
WorkoutStep* steps = programs[selected_program].steps; //Nastavi kazalec? (pointer)
na mesto v pomnilniku, kjer se nahaja steps

for(int i = 0; i < stepCount; i++){
    excecuteStep(steps[i], i); //Izvede korak
    delay(steps[i].delay * 1000); //Pocaka bastavljen cas
}
//Meni po koncanem programu
lcd.clear();
lcd.setCursor(1, 0);
lcd.print(programs[selected_program].name);
lcd.setCursor(6, 1);
lcd.print("Zaključen");
lcd.setCursor(0, 2);
lcd.print("<OK> Izberi program");
lcd.setCursor(0, 2);
lcd.print("<UP> <DOWN> Osvezi");
while(digitalRead(OK_BTN) || (digitalRead(UP_BTN) && digitalRead(DWN_BTN)));
if(!digitalRead(UP_BTN) && !digitalRead(DWN_BTN)){
    getPrograms();
}
if(!digitalRead(OK_BTN)){
    selectPrpgramToRun();
}
}

// ===== SETUP =====

void setup() {
    //Nastavi serijsko komunikacijo
    Serial.begin(115200);
    delay(1000);

    //Priprvi LCD prikazovalnik
    lcd.init();
```

```
lcd.backlight();

//Nastavi funkcije priključkov
pinMode(OK_BTN, INPUT_PULLUP);
pinMode(UP_BTN, INPUT_PULLUP);
pinMode(DWN_BTN, INPUT_PULLUP);

pinMode(LEFT_DAT_OUT, OUTPUT);
pinMode(LEFT_SER_OUT, OUTPUT);
pinMode(LEFT_LATCH_OUT, OUTPUT);

pinMode(RIGHT_DAT_OUT, OUTPUT);
pinMode(RIGHT_LATCH_OUT, OUTPUT);
pinMode(RIGHT_SER_OUT, OUTPUT);

pinMode(LEFT_CLOCK_IN, INPUT);
pinMode(LEFT_DATA_IN, INPUT);
pinMode(LEFT_LATCH_IN, INPUT);

pinMode(RIGHT_CLOCK_IN, INPUT);
pinMode(RIGHT_DATA_IN, INPUT);
pinMode(RIGHT_LATCH_IN, INPUT);

//Povezava na WiFi
WiFi.begin("", "");
#ifdef DEBUG
    Serial.print("Povezovanje na WiFi");
#endif
//Izpis podatkov
lcd.setCursor(0, 0);
lcd.print("Vzpostavljam povezavo");

for(int i = 0; i < MWCT; i++){
    if(WiFi.status() == WL_CONNECTED){
        #ifdef DEBUG
            Serial.println("Povezava vzpostavljena");
        #endif
    }
}
```

```
        lcd.setCursor(1, 1);
        lcd.print("Povezano!");
    }
}

//Statis povezave WiFi
#ifdef DEBUG
    Serial.println("\nPovezano!");
    Serial.println("IP: " + WiFi.localIP().toString());
#endif
lcd.setCursor(1, 0);
lcd.print("Povezano");

getPrograms();
}

void loop() {
    //Neskoncno izvajanje programa
    selectPrpgramToRun();
    while(digitalRead(OK_BTN));
    while(!digitalRead(OK_BTN));
}
```

## 12.2. PROGRAM ZA STREŽNIK

### 12.2.B. Priloga B - add\_program.html

```
<!DOCTYPE html> //Glava HTML dokumenta
<html lang="si"> //Jezik spletne strani
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ustvari plezalni program</title> //Ime spletne strani
</head>
<link rel="stylesheet" href="style.css">
<body>
//Obrazec za plezalni program
  <form action="ad_prog_sy.php", method="post">
    <input type="text" name="prog_name" placeholder="Ime programa">
    <input type="number" name="wrkt_n" placeholder="Stevilo vadb">
    <input type="submit" value="Ustvari">
  </form>

</body>
</html>
```

## 12.2.C. Priloga C - get\_wrkt.php

```
<?php
header('Content-Type: application/json');
require_once("db.php"); //Povezava z bazo podatkov

$tablesResult = $con->query("SHOW TABLES"); //Pridobi imena tabel
$allData = [];

//Shrani vsebino tabel v .json datoteko
if ($tablesResult->num_rows > 0) {
    while ($tableRow = $tablesResult->fetch_array()) {
        $tableName = $tableRow[0]; // ime tabele
        $sql = "SELECT * FROM `{$tableName}`";
        $result = $con->query($sql);

        $rows = [];
        if ($result && $result->num_rows > 0) {
            while ($row = $result->fetch_assoc()) {
                $rowInt = [];
                foreach ($row as $key => $value) {
                    if (is_numeric($value)) {
                        $rowInt[$key] = (int)$value;
                    } else {
                        $rowInt[$key] = $value;
                    }
                }
                $rows[] = $rowInt;
            }
        }
        $allData[$tableName] = $rows;
    }
}
echo json_encode($allData, JSON_PRETTY_PRINT); //Vrne .json datoteko
$con->close(); //Zapre povezavo z zbirko podatkov
?>
```

## 12.2.D. Priloga D - save\_prog.php

```
<?php
$success = true;
$error_message = "";

if(!isset($_POST['wrkt_n']) || !isset($_POST['prog_name'])){
    $success = false;
    $error_message = "Manjkajo podatki.";
} else {

    $wrkt_n = $_POST['wrkt_n'];
    $prog_name = $_POST['prog_name'];

    require_once("db.php");

    for ($i = 0; $i < $wrkt_n; $i++) {

        $grip_l = $_POST["grip_l".$i] ?? 0;
        $grip_r = $_POST["grip_r".$i] ?? 0;
        $balance = $_POST["balance".$i] ?? 0;
        $time = $_POST["time".$i] ?? 0;
        $delay = $_POST["delay".$i] ?? 0;

        $query = "INSERT INTO `".$prog_name.`
            (num, grip_l, grip_r, balance, time, delay)
            VALUES
            ($i, $grip_l, $grip_r, $balance, $time, $delay)";

        if(!mysqli_query($con, $query)){
            $success = false;
            $error_message = mysqli_error($con);
            break;
        }
    }
}
?>
```

```
<!DOCTYPE html>
<html lang="sl">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Status shranjevanja</title>

<style>
body {
  min-height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  background: linear-gradient(135deg, #1e3c72, #2a5298);
  font-family: "Segoe UI", sans-serif;
}

.card {
  background: white;
  padding: 40px;
  border-radius: 20px;
  box-shadow: 0 20px 40px rgba(0,0,0,0.2);
  text-align: center;
  max-width: 400px;
  width: 90%;
}

.success {
  color: #2e7d32;
  font-size: 22px;
  font-weight: 600;
}

.error {
  color: #c62828;
  font-size: 18px;
```

```
font-weight: 600;
}

button {
margin-top: 25px;
padding: 12px 20px;
border: none;
border-radius: 10px;
background: linear-gradient(135deg, #2a5298, #1e3c72);
color: white;
font-size: 15px;
cursor: pointer;
transition: 0.3s;
}

button:hover {
transform: translateY(-3px);
box-shadow: 0 8px 20px rgba(0,0,0,0.2);
}
</style>

</head>
<body>

<div class="card">

<?php if($success): ?>
<div class="success">
 Program <br><strong><?php echo htmlspecialchars($prog_name); ?></strong><br>
je bil uspešno shranjen!
</div>
<?php else: ?>
<div class="error">
 Napaka pri shranjevanju:<br>
<?php echo htmlspecialchars($error_message); ?>
</div>
<?php endif: ?>
```

```
<button onclick="window.location.href='add_program.html'">
```

```
  Nazaj
```

```
</button>
```

```
</div>
```

```
</body>
```

```
</html>
```

## 12.1.E. Priloga E - add\_prog\_sy.php

```

<?php
    //Pridobi podatke
    if(!isset($_POST['prog_name'])){
        exit("Ni veljavnega imena");
    }
    $prog_name = $_POST['prog_name'];
    $wrkt_n = $_POST['wrkt_n'];

    require_once("db.php"); //Povezava z zbirko podatkov

    //Ustvari novo tabelo v zbirki podatkov
    $query = "CREATE TABLE `\$prog_name` (
        `num` tinyint(4) NOT NULL,
        `grip_l` tinyint(4) NOT NULL,
        `grip_r` tinyint(4) NOT NULL,
        `balanse` tinyint(4) NOT NULL,
        `time` tinyint(4) DEFAULT NULL,
        `delay` tinyint(4) DEFAULT NULL
    )";
    mysqli_query($con, $query);

?>

<!DOCTYPE html> //Glava HTML dokumenta
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ustvari nov program</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
//Ustvari vpisna polja za nastavitve korakov
<form action="save_prog.php" method="post">
    <?php
        for($i = 0; $i < $wrkt_n; $i++){
            $grip_name_l = "grip_l" . $i;
            $grip_name_r = "grip_r" . $i;
            $balance_name = "balance" . $i;

```

```

$time_name = "time" . $i;
$delay_name = "delay" . $i;
$title = "Podprogram " . $i;
?>
<h2><?php echo $title ?></h2>
<input type="radio" name="<?= $grip_name_1 ?>" value="1" class="required-field">
<input type="radio" name="<?= $grip_name_1 ?>" value="2" class="required-field">
<input type="radio" name="<?= $grip_name_1 ?>" value="3" class="required-field">
<input type="radio" name="<?= $grip_name_1 ?>" value="4" class="required-field">
<input type="radio" name="<?= $grip_name_1 ?>" value="5" class="required-field">
<br>
<input type="radio" name="<?= $grip_name_r ?>" value="1" class="required-field">
<input type="radio" name="<?= $grip_name_r ?>" value="2" class="required-field">
<input type="radio" name="<?= $grip_name_r ?>" value="3" class="required-field">
<input type="radio" name="<?= $grip_name_r ?>" value="4" class="required-field">
<input type="radio" name="<?= $grip_name_r ?>" value="5" class="required-field">
<br>
<input type="range" name="<?php echo $balance_name; ?>" min="0" max="100" placeholder="Balans"
class="required-field">
<br>
<input type="number" name="<?php echo $time_name; ?>" placeholder="ĀŠas strajanja [s]" class="required-
field">
<br>
<input type="number" name="<?php echo $delay_name; ?>" placeholder="Pavza [s]" class="required-field">
<br>
<br>
<br>
<?php
}
?>
<input type="hidden" name="wrkt_n" value="<?php echo $wrkt_n; ?>">
<input type="hidden" name="prog_name" value="<?php echo $prog_name ?>">
<input type="submit" value="Sharni" id="submit">

</form>
//Onemogoči oddajo programa, če niso vpisani vsi podatki
<script>
document.addEventListener("DOMContentLoaded", function() {
const submitBtn = document.getElementById("submit");
const requiredInputs = document.querySelectorAll(".required-field");

function checkInputs() {

```

```
    let allFilled = true;
    requiredInputs.forEach(input => {
      if (input.value.trim() === "") {
        allFilled = false;
      }
    });
    submitBtn.disabled = !allFilled;
  }

  requiredInputs.forEach(input => {
    input.addEventListener("input", checkInputs);
  });

  checkInputs();
});
</script>

</body>
</html>
```

## 12.2.F. Priloga F - db.php

```
<?php
$servername = "";
$username = "";
$password = "";
$dbname = "plezarija";

//Vzpostavi povezavo z zbirko podatkov
//Zgornje vrednosti so izbrisane zaradi podatkov o mojem strežniku
$con = new mysqli($servername, $username, $password, $dbname);

// Preveri povezavo
if ($con->connect_error) {
    http_response_code(500);
    echo json_encode(["error" => "connection failed"]);
    exit();
}
?>
```

## 12.2.G. Priloga G - Struktura SQL baze podatkov

//SQL ukaz za ustvaritev tabele, v katerem je prikazana struktura tabele

```
CREATE TABLE `IME PROGRAMA` (
```

```
  `num` tinyint(4) NOT NULL,
```

```
  `grip_l` tinyint(4) NOT NULL,
```

```
  `grip_r` tinyint(4) NOT NULL,
```

```
  `balanse` tinyint(4) NOT NULL,
```

```
  `time` tinyint(4) DEFAULT NULL,
```

```
  `delay` tinyint(4) DEFAULT NULL
```

```
)
```