

Upravljanje rukama humanoidnog robota TIOSS

Brčić, Ivan

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:280094>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 133

UPRAVLJANJE RUKAMA HUMANOIDNOG ROBOTA TIOSS

Ivan Brčić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 133

UPRAVLJANJE RUKAMA HUMANOIDNOG ROBOTA TIOSS

Ivan Brčić

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 133

Pristupnik: **Ivan Brčić (0036523401)**
Studij: Elektrotehnika i informacijska tehnologija
Profil: Elektroničko i računalno inženjerstvo
Mentorica: izv. prof. dr. sc. Ana Sović Kržić

Zadatak: **Upravljanje rukama humanoidnog robota TIOSS**

Opis zadatka:

TIOSS je prvi humanoidni robot koji je šetao Zagrebom 1961. godine, a ime mu je kratica za teledirigirani izvršni organ samoorganizirajućeg sustava. Radi se o robotu željezne konstrukcije, visokom preko dva metra i teškom preko 150 kg. U okviru diplomskog rada je potrebno izvršiti modernizaciju robota s naglaskom na rukama i nogama. Potrebno je istražiti originalno upravljanje rukama i hvataljkama na vrhu ruku. Osmisliti i implementirati povezivanje elektronike ruku s Raspberry Pi mikroračunalom. Omogućiti udaljeno upravljanje rukama robota. Istražiti mogućnost glasovnog upravljanja robotom. Implementirati upravljanje rukama pomoću glasa.

Rok za predaju rada: 28. lipnja 2024.

Zahvala

Prvenstveno bih se zahvalio mojoj mentorici, izv. prof. dr. sc. Ani Sović Kržić, koja mi je ponudila ovu temu te mi omogućila da radim na njoj slobodno, bez pritiska i uz njenu podršku. Također, bila je prepuna razumijevanja i motivacije kroz čitav studij.

Osoba koja me pratila i pomagala mi kroz čitav period rada na ovom zadatku je mr. sc. Marijan Kuri. Omogućio mi je da uz njega radim, učim, propitkujem i razmišljam bez da mi je ikada bilo neugodno ili stresno. Uložio je izuzetno puno vremena i živaca, pokazao je strpljenje i volju, a ja mu se na svemu tome mogu samo neizmjereno zahvaliti.

Smatram da je potrebno zahvaliti se i mojim kolegama i bliskim prijateljima koje sam stekao kroz studij, ali i van njega. Često su bili tu kada je bilo potrebno, pružili mi podršku ili jednostavno proveli vrijeme sa mnom.

Za kraj, hvala mojim roditeljima i bližnjima koji su uvijek bili moj uzor, moja motivacija, podrška, ali i moji najveći, uvijek potrebni, kritičari.

Sadržaj

1. Uvod.....	4
2. Robot TIOSS	5
2.1. Restauracija motora	6
2.2. Testiranje i funkcionalnost motora	7
3. Projektiranje upravljačkog sklopa	9
3.1. Problematika i početna ideja	9
3.2. Razrada ideje za upravljački sklop	10
3.3. Senzor položaja	16
3.4. Mehaničke sklopke za ograničavanje kretnje ruku i glave	20
3.5. <i>Raspberry Pi</i> računalo.....	21
3.6. Priklučci na upravljačkom sklopu	26
4. Izrada Pločice za upravljački sklop	27
4.1. Kompletna električna shema	27
4.2. Priprema za PCB dizajn.....	31
4.3. PCB dizajn	32
4.4. Izrada PCB pločice	35
5. Programska podrška za upravljanje robotom	38
5.1. Povezivanje GPIO priključaka s upravljačkim sklopom	39
5.2. Opis biblioteka korištenih u programskom kodu	40
5.3. Programski kod.....	41
5.3.1. Uvoz potrebnih biblioteka.....	41
5.3.2. Inicijalizacija GPIO priključaka	41
5.3.3. Definiranje početnih stanja GPIO priključaka	42
5.3.4. Postavljanje kanala ADP-a	42
5.3.5. Pretvorba vrijednosti ADP-a.....	43
5.3.6. Glavni dio programa	43
5.3.7. Algoritam za samokalibraciju položaja ruku i glave	44
6. Ispitivanje ispravnosti rada.....	46
6.1. Nezgoda pri ispitivanju sklopa.....	46
6.2. Testiranje sklopa na robotu TIOSS	47
7. Moguća unaprjeđenja	49
8. Zaključak.....	50

9. Literatura.....	51
--------------------	----

1. Uvod

Kako bi ovaj rad dobio kontekst i smisao potrebno je vratiti se u prošlost, točnije, u šezdesete godine 20. stoljeća. U tom je razdoblju svjetlo dana ugledao prvi humanoidni robot u Hrvatskoj – TIOSS, čije ime predstavlja skraćenicu za teledirigirani izvršni organ samoorganizirajućeg sustava. U to vrijeme, robot je bio prava atrakcija pa je tako, primjerice, dijelio letke na Zagrebačkom velesajmu, a 1968. godine pojavio se i na Trgu bana Josipa Jelačića i time privukao veliku pažnju i zanimanje građana [1].

Sama ideja o stvaranju robota nastala je još 1956. godine kada je Branimir Makanec razvijao interes za to područje, a računala su bila veličine zgrada i ne svima dostupna. Ipak, Branimir Makanec osnovao je grupu mladih kibernetičara koji su uspjeli izraditi robota TIOSS i to koristeći otpad poput dijelova američkih aviona iz Drugog svjetskog rata i motora s automobila. S obzirom da računala nisu bila dostupna kao danas, koristeći telefonske releje, napravio je računalo kojim se upravljalo funkcijama robota, a pokretali su ga akumulatori koji su bili povezani na dva jaka motora u nogama robota. Ti motori su naizmjenično pomicali jednu po jednu nogu robota kako bi se simulirao ljudski hod, a mogao je postići i brzinu od 30 kilometara na sat. Robot je imponirao svojom veličinom jer je bio visok 220 cm i težio je više od 150 kg. Uz to, uspjeli su ostvariti i komunikaciju s robotom tako što je osoba koja upravlja robotom slušala pitanja ljudi preko mikrofona te im odgovarala, a odgovor se čuo preko zvučnika u glavi robota. Osim zvučnika, glava robota je bila opremljena i s optičkim sensorima koji su reagirali na promjenu svjetlosti kako bi robot okretao glavu prema najvećem svjetlosnom podražaju [2].

Kroz vrijeme, robot se selio po raznim ustanovama, koristio u različite svrhe, a s obzirom da se o njemu nije brinulo na primjeren način, dijelovi robota su izvađeni, pokradeni ili uništeni. Tako su nestali akumulatori, motori i releji, a naposljetku je darovan jednom nastavniku u osnovnoj školi. Na dug period vremena robotu se izgubio trag, a robota je više od 20 godina čuvao Mario Stipandić koji je s Klubom mladih tehničara Dubrave pokušao rekonstruirati robota, ali neuspješno [2]. Prije četiri godine Branimir Makanec se ponovno susreo sa svojom kreacijom povodom 88. rođendana, a onda ga je i darovao izv. prof. dr. sc. Ani Sović Kržić koja je pokrenula proces rekonstrukcije i modernizacije robota kako bi ga se vratilo u život.

Kolege s Fakulteta elektrotehnike i računarstva radili su na modernizaciji kretanja robota TIOSS te korištenju robotskog vida i govora. Ovaj rad se fokusira na rekonstrukciju i modernizaciju upravljanja kretanjem autonomnih ruku s hvataljkama. Za upravljanje sklopom koristi se *Raspberry Pi* računalo pri čemu je projektirani upravljački sklop kompatibilan s različitim inačicama računala uz prilagodbu programskog koda koji se koristi za upravljanje robotom. Također se omogućuje praćenje položaja ruku i glave korištenjem senzora položaja koji su postavljeni na primjerene pozicije. Ovime se postavljaju temelji za dodatna unaprjeđenja robota TIOSS poput upravljanja robotom glasovnim naredbama ili korištenje računalnog vida za imitaciju ljudskih pokreta ruku.

2. Robot TIOSS

Humanoidni robot TIOSS prvi je robot u Hrvatskoj te je, za vrijeme u kojem se pojavio, bio zaista impresivna pojava. Težak više od 150 kg i visok 220 cm, daleko je od lagane i jednostavno pomične konstrukcije. Humanoidne značajke ovog robota primjećuju se u fizionomiji robota koji ima dvije noge, dvije ruke, hvataljke koje imitiraju šake, odnosno prste i glavu s očima. Optički senzori, koji su zapravo služili funkciju očiju, nalazili su se na bočnim stranama glave kako bi se glava okretala prema najsnažnijem svjetlosnom podražaju. Izvorno je robot mogao pomicati jednu po jednu nogu i na taj način simulirati ljudski hod, a nakon restauracije motora nogu robot TIOSS ponovno se mogao kretati vlastitim motorima [3]. U svakoj nozi nalazi se jedan 12-voltni akumulator koji su serijski spojeni kako bi zajedno davali izvor od 24V. Taj dio je postavio dobre temelje za daljnji rad na robotu jer je napajanje već osigurano.



Slika 2.1. Robot TIOSS

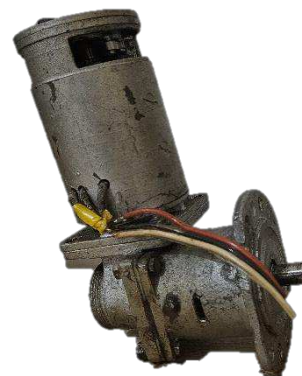
U ovoj fazi rada na robotu TIOSS, ne samo u ovom radu nego i u drugim radovima, radilo se na restauraciji motora ruku i hvataljki, dizajniranju i izradi upravljačkog sklopa za upravljanje glavom i rukama, daljinskom upravljanju robotom pomoću

kontrolera te na interakciji robota s ljudima. Ta interakcija bazira se na prepoznavanju ljudskih emocija te na verbalnoj komunikaciji s ljudima.

2.1. Restauracija motora

S obzirom da je početna faza restauracije robota već prethodno napravljena, u ovom radu se fokusiralo na restauraciju dvaju motora koji su zaduženi za pokretanje autonomnih ruku robota te restauraciju hvataljki ruku. Motori godinama nisu skidani sa robota, a nisu ni korišteni te je prvi zadatak bio skinuti motore sa robota i procijeniti u kojem su zapravo stanju.

Robotu TIOSS (Slika 2.1.) se s unutrašnje strane, u razini ramena, nalaze motori koji pokreću ruke te se na njih nadovezuje hvataljka koja imitira ljudsku šaku. Nakon što su su motori i ruke skinuti s robota, bilo je vidljivo kako nisu u najboljem stanju (Slika 2.2.). Također, bilo je jasno da je na motore, u nekom trenutku, bio nanesen srebrni sloj koji je otežao raspoznavanje i vidljivost dijelova motora. Kada se taj srebrni sloj maknuo, stvari su postale jasnije te je bilo jednostavnije shvatiti od kuda krenuti. Mast unutar motora se skorila te su dijelovi motora bili teško razdvojivi, a unutar



Slika 2.2. Motor ruke

motora se nakupilo i dosta nečistoće. Tri žice koje su povezane na namote motora su bile loše povezane, krute i omotane samo trakom za izolaciju te su se morale zamijeniti novim žicama (Slika 2.3.) koje su spojene na izvode namota u motorima i zaštićene izolacijom. Potrebno je bilo očistiti unutrašnjost motora od nakupljene zakorjele masti i ostalih nečistoća te ponovno podmazati zupčanike motora (Slika 2.4.) korištenjem sredstava za odmaščivanje, čišćenje i podmazivanje. Četkice motora koje su se nalazile unutar motora nisu više bile u iskoristivom stanju te je bilo potrebno u motore ugraditi nove. Napravljene su nove grafitne četkice (Slika 2.5.) te su postavljene u motore. Bilo je potrebno doraditi još nekoliko detalja na motorima poput mjedenih ležajeva koji su dodani na osovinu motora na koju se spajaju ruke robota.



Slika 2.3. Nove žice na motoru



Slika 2.4. Podmazani zupčanci



Slika 2.5. Nove četkice

Hvataljke je ipak bilo nešto lakše osposobiti za ponovnu upotrebu jer nisu bile toliko izložene vanjskim utjecajima budući da su bile unutar ruku robota. Hvataljke se upravljaju na principu solenoida gdje jedna hvataljka stišće, a druga pušta uslijed prolaska struje. Ipak, promijenjene su žice koje su povezane na namote koji vode do hvataljki te su izvučene u unutrašnjost robota zajedno s žicama iz motora ruku kako bi se jednostavnije povezale na sklop za upravljanje koji je kasnije projektiran i napravljen.

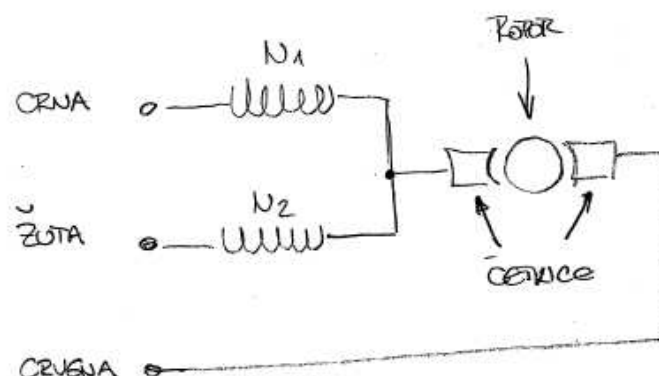
Motor glave, koji nije bio fokus ovoga rada, prethodno je već restauriran i osposobljen za rad, ali tijekom rada na robotu došlo je do manjih oštećenja pa se i na njemu moralo napraviti nekoliko popravaka kako bi ispravno funkcionirao. Ovdje je napomenut jer će se motorom glave, na gotovo identičan način, upravljati preko istog upravljačkog sklopa te će se i na njega postaviti senzor položaja.

U ovom procesu restauracije i osposobljavanja motora ruku i glave te hvataljki robota potrebno je naglasiti važnost i ulogu mr. sc. Marijana Kurija koji je uložio puno vremena i truda u ovaj dio posla te ga odradio besprijekorno te tako omogućio uspješnost predmetnog procesa.

2.2. Testiranje i funkcionalnost motora

Nakon restauracije i osposobljavanja motora, sljedeći korak bio je testiranje funkcionalnosti motora. Kao što je prethodno navedeno, iz motora izlaze tri žice: crvena, crna i žuta. Naime, radi se o elektromotoru koji radi kada su rotor i namot spojeni serijski (Slika 2.6.), a smjer vrtnje ovisi o namotu koji je spojen u seriju s

rotorom. Spajanje oba namota uzrokuje poništavanje magnetskog polja i elektromotor ne može raditi, a dvostruka struja kroz četkice i rotor se ne preporuča.



Slika 2.6. Princip rada elektromotora

Crvena žica spaja se na pozitivni kraj napajanja od 24V, a spajanjem crne žice na negativni kraj napajanja elektromotor rotira u lijevo dok se spajanjem žute žice na negativni kraj napajanja elektromotor rotira u desno.

Iz navedenog je vidljivo da princip rada elektromotora nije pretjerano kompleksan, a princip rada hvataljki je još jednostavniji. Naime, hvataljke funkcioniraju na principu solenoida (elektromehanički uređaj za pretvaranje električne energije u mehanički pokret). Solenoid čini zavojnica (cilindrični namotana bakrena žica) i pomični klip unutar te zavojnice. Uslijed prolaska struje kroz zavojnicu stvara se magnetsko polje koje privlači ili odbija pomični klip. Taj pomak klipa, koji je povezan s hvataljkom, uzrokuje njeno otvaranje ili zatvaranje, ovisno o dizajnu. Kod hvataljke nema izmjene polariteta nego je jedna žica konstantno na pozitivnom kraju napajanja dok se druga spaja ili odspaja s negativnog kraja napajanja kako bi se hvataljka zatvorila ili otvorila.

S obzirom da će se motorom glave također upravljati preko istog upravljačkog sklopa kojim će se upravljati motorima ruku i hvataljkama, potrebno je napomenuti da je funkcionalnost motora glave istovjetna funkcionalnosti motora ruku. U dizajnu upravljačkog sklopa će se motoru glave pristupati na isti način kao i motorima ruku.

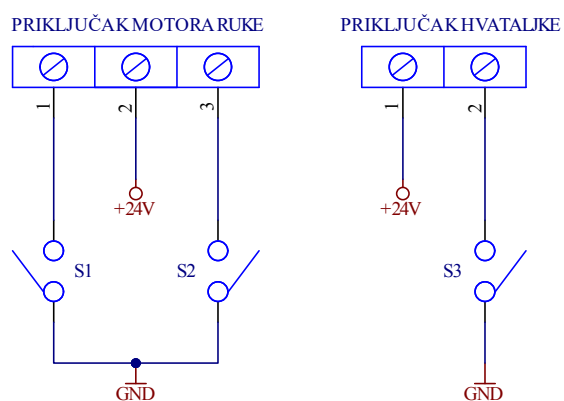
3. Projektiranje upravljačkog sklopa

U ovom dijelu rada dati će se detaljan pregled cjelokupnog procesa projektiranja i dizajniranja upravljačkog sklopa za upravljanje hvataljkama te motorima ruku i glave. Za početak, objasnit će se problematika i prvotna ideja za dizajn upravljačkog sklopa. Za softverski, programabilni dio upravljanja robotom će se koristiti *Raspberry Pi* računalo, a inačice računala korištene u ovom radu su *Raspberry Pi 5* i *Raspberry Pi 3*. Tijekom dizajniranja nekog rješenja problema često su se pojavile poteškoće ili bolje i drugačije ideje koje zahtijevaju male preinake ili kompletnu promjenu pristupa. Radi pojednostavljenja prikaza u nadolazećim poglavljima opisan je proces projektiranja upravljačkog sklopa za jedan motor ruke i za jednu hvataljku, a u kasnijem, konačnom dizajnu će se opisani sklop samo ponoviti za drugu ruku, drugu hvataljku i za glavu robota.

3.1. Problematika i početna ideja

S obzirom na prethodno opisan način rada elektromotora i hvataljki potrebno je osmisлити upravljački sklop kojim će se moći upravljati rukama i hvataljkama robota TIOSS. Taj upravljački sklop trebao bi omogućiti da se jednostavno i brzo izmjenjuje smjer kretanja ruku i glave te da se hvataljke mogu stiskati, odnosno puštati. Kako ne bi došlo do oštećenja robota bitno je da se kretanje ruku robota ograniči na određenu dinamiku s obzirom na mehanička ograničenja zglobova.

Jasno je da će temelj upravljačkog sklopa biti upravljanje sklopkama kojima će se zatvarati strujni krugovi i pokretati motori, odnosno hvataljka. Izvorna, pojednostavljena ideja (Slika 3.1.) bila je da se jedna od žica motora fiksno postavi na pozitivni kraj napajanja od 24V, a da se s dvjema sklopkama upravlja i ovisno o tome koja je od sklopki zatvorena, a koja otvorena, jedna od dvije preostale žice motora spoji na negativan kraj napajanja te se time zatvara strujni krug, a elektromotor rotira. Hvataljkama je jednostavnije upravljati, jer je jedna žica



Slika 3.1. Početna ideja za upravljački sklop

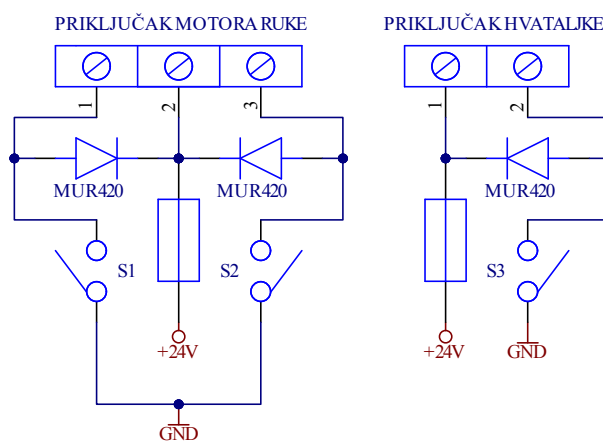
konstantno na pozitivnom kraju napajanja od 24V, a drugi kraj se preko upravljive sklopke spaja na negativni kraj napajanja.

3.2. Razrada ideje za upravljački sklop

Nakon osmišljanja osnovne ideje za upravljački sklop, pojavila su se očigledna i logična pitanja na koja je bilo potrebno dati odgovore.

Prvi problem za koji je trebalo pronaći rješenje je stabilnost i sigurnost upravljačkog sklopa. U paralelu s teretom potrebno je postaviti zaštitne diode kako bi se zaštitilo od prenapona se osiguralo da ne dođe do provođenja struje u suprotnom smjeru. Tip zaštitne diode koja se koristila u ovom sklopu je poluvodička dioda MUR420 [5]

koju karakterizira izuzetno brzo vrijeme oporavka koje je tipično oko 25ns, ali i maksimalni povratni napon od 200V. Također je potrebno postaviti osigurač na pozitivni dio napajanja elektromotora tako da u slučaju neke nepredviđene situacije ne dođe do većih neželjenih oštećenja sklopa ili samog elektromotora. Osigurači korišteni u ovom radu su električni osigurači

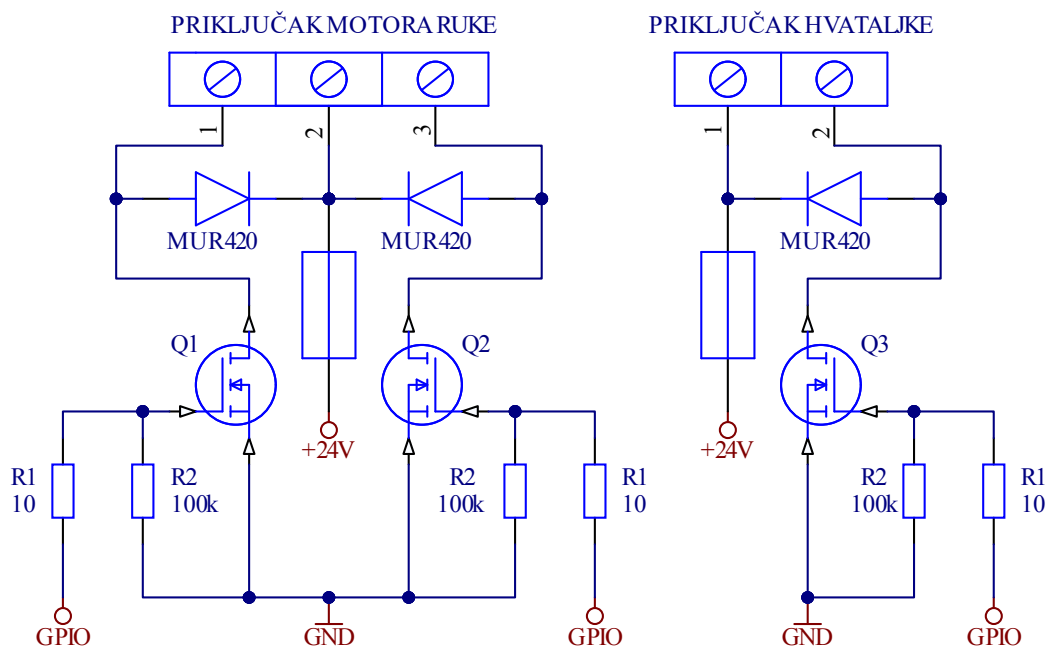


Slika 3.2. Dodavanje osigurača i zaštitnih dioda

cilindričnog oblika i staklenog kućišta kako bi se omogućila vizualna inspekcija tanke žice unutar osigurača. Ideja je umetnuti ih u držače kako bi montiranje i zamjena bila što jednostavnija. Korištene su dvije vrste osigurača, jedan osigurava do 5A i on se koristi kod priključka za povezivanje hvataljki, dok drugi osigurava do 6,3A i koristi se kod priključaka za motore ruku.

Sljedeći korak bio je riješiti pitanje upravljivih sklopki. Poznato je da su tranzistori često korišteni upravo kao sklopke, a tako je i u ovom radu. Odabran je N-kanalni MOSFET IRLZ44N [6] kojeg karakterizira lako upravljanje zbog niskog napona praga, niski otpor u uključenom stanju i visoka struja odvoda. Također dolazi u robusnom pakiranju što omogućava učinkovitu disipaciju topline te ga je lako postaviti na hladnjak kako bi se disipacija topline dodatno povećala. Uvod (eng.

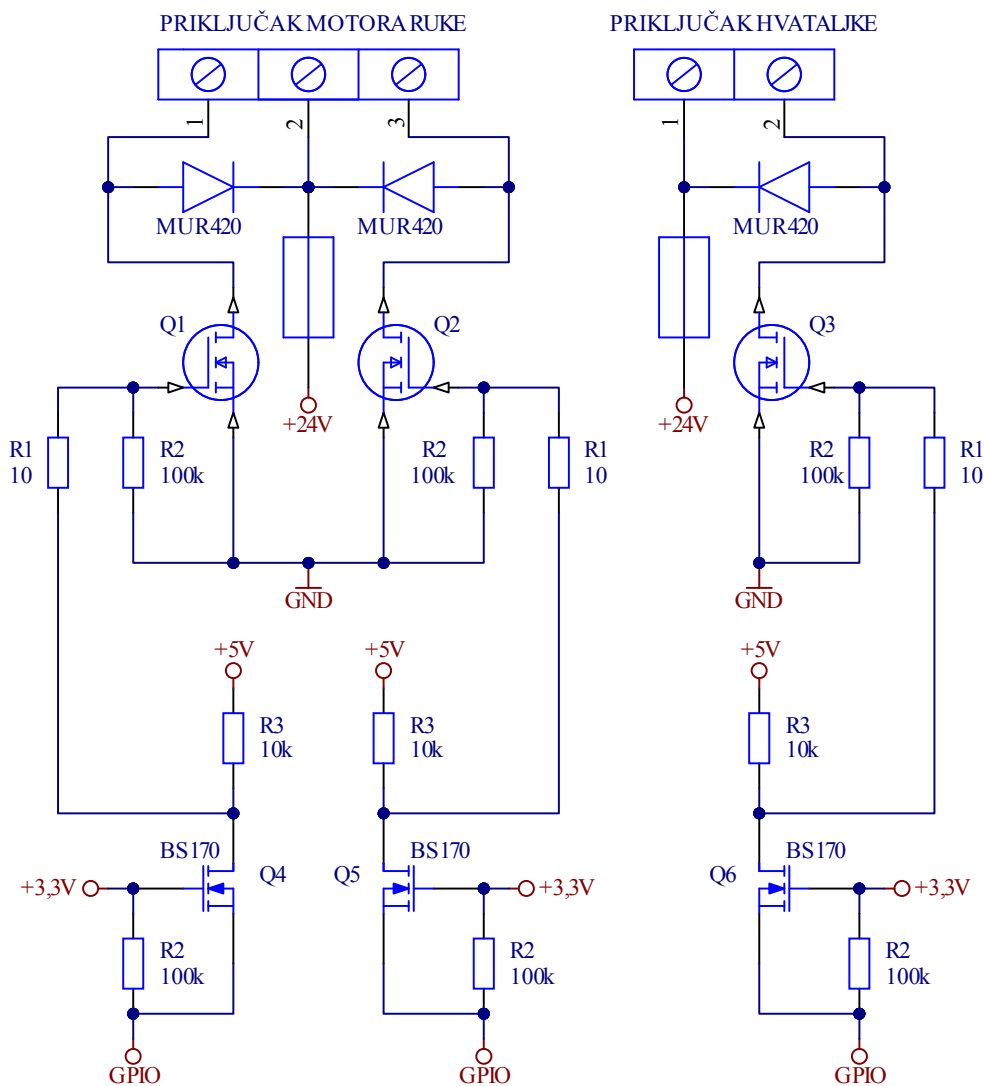
Source) MOSFET-a spojen je na anodu diode, odvod (eng. Drain) je spojen na masu odnosno na negativni kraj napajanja, a upravljačka elektroda (eng. Gate) ima spojen veliki otpor prema masi, reda veličine $100k\Omega$, a upravljačkom elektrodom upravlja se pomoću GPIO priključka s *Raspberry Pi* računala [7]. Između GPIO priključka i upravljačke elektrode postavljen je maleni otpor, reda veličine 10Ω iako nije nužno da bude postavljen (Slika 3.3.).



Slika 3.3. Dodavanje MOSFET-a

Raspberry Pi na izlazima GPIO priključaka daje dvije naponske razine. Niska razina na izlazu daje napon od 0V, a visoka razina na izlazu pinova daje 3,3V. Taj napon visoke razine vjerojatno bi bio dovoljan da se MOSFET-i uključe, odnosno da provede struju od uvida prema odvodu te da se time zatvori strujni krug, ali također se ne bi smjeli zanemariti nepredviđeni slučajevi. Naponi koje daju GPIO priključci relativno su blizu naponu praga MOSFETA pa se tako ne može isključiti mogućnost da uslijed nekog nepredviđenog pada napona u sklopu dođe do situacije da na upravljačku elektrodu dođe premaleni napon. Kada bi trošilo bilo samo LED dioda (Light Emitting Diode), to bi predstavljalo manji problem, ali trošilo s kojim se upravlja u ovom radu su motori ruku robota TIOSS koji je poprilično velik i potrebno je osigurati što sigurniji rad i upravljanje robotom.

Kako bi se osiguralo da na upravljačku elektrodu dođe dovoljno visok napon odlučeno je postaviti visoku razinu u 5V umjesto napona od 3,3V koji izvorno daju GPIO priključci *Raspberry Pi* računala. To se može postići postavljanjem još jednog MOSFET-a kojem će uvod preko otpornika reda veličine 10kΩ biti povezan s naponom od 5V, a istovremeno će biti spojen na upravljačku elektrodu MOSFET-a postavljenih u prethodnom koraku. Također, upravljačka elektroda ovih dodatnih MOSFET-a će biti na konstantnom potencijalu od 3,3V, a s odvodom će biti spojena preko velikog otpornika reda veličine 100kΩ. Dodavanjem dodatnih MOSFET-a, GPIO priključci su sada povezani na odvođe tih tranzistora (Slika 3.4.). Tranzistori odabrani za ovu funkciju su BS170 [8] N-kanalni MOSFET-i koji su jeftini, pogodni za male struje te se lako montiraju na tiskane pločice.



Slika 3.4. Dodatna razina MOSFET-a

Sada kada su dodani BS170 MOSFET-i, razlikujemo dva slučaja:

1) *GPIO priključak je u visokoj razini*

Kada je na izlazu GPIO priključka visoka razina, napon odvoda MOSFET-a iznosi 3,3V te je napon između upravljačke elektrode i uvoda jednak nuli s obzirom da je napon na upravljačkoj elektrodi fiksiran na 3,3V. Time MOSFET nije uključen te je odvod tranzistora na 5V i taj napon dolazi na upravljačku elektrodu sljedećeg MOSFET-a

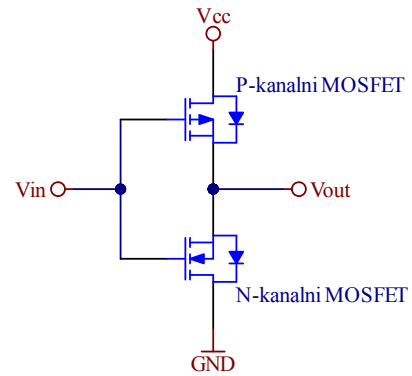
2) *GPIO priključak je u niskoj razini*

Kada je na izlazu GPIO priključka niska razina, napon odvoda MOSFET-a iznosi 0V te je napon između upravljačke elektrode i uvoda jednak 3,3V što znači da će MOSFET biti uključen, a odvod će biti na 0V. Time na upravljačku elektrodu sljedećeg MOSFET-a također dolazi 0V.

Ovim proširenjem upravljačkog sklopa osigurano je da na upravljačku elektrodu MOSFET-a kojima se zatvara strujni krug i upravlja trošilom dolazi dovoljan napon kako bismo smanjili vjerojatnost da tranzistor ne provede ako bi na upravljačku elektrodu došao premali napon.

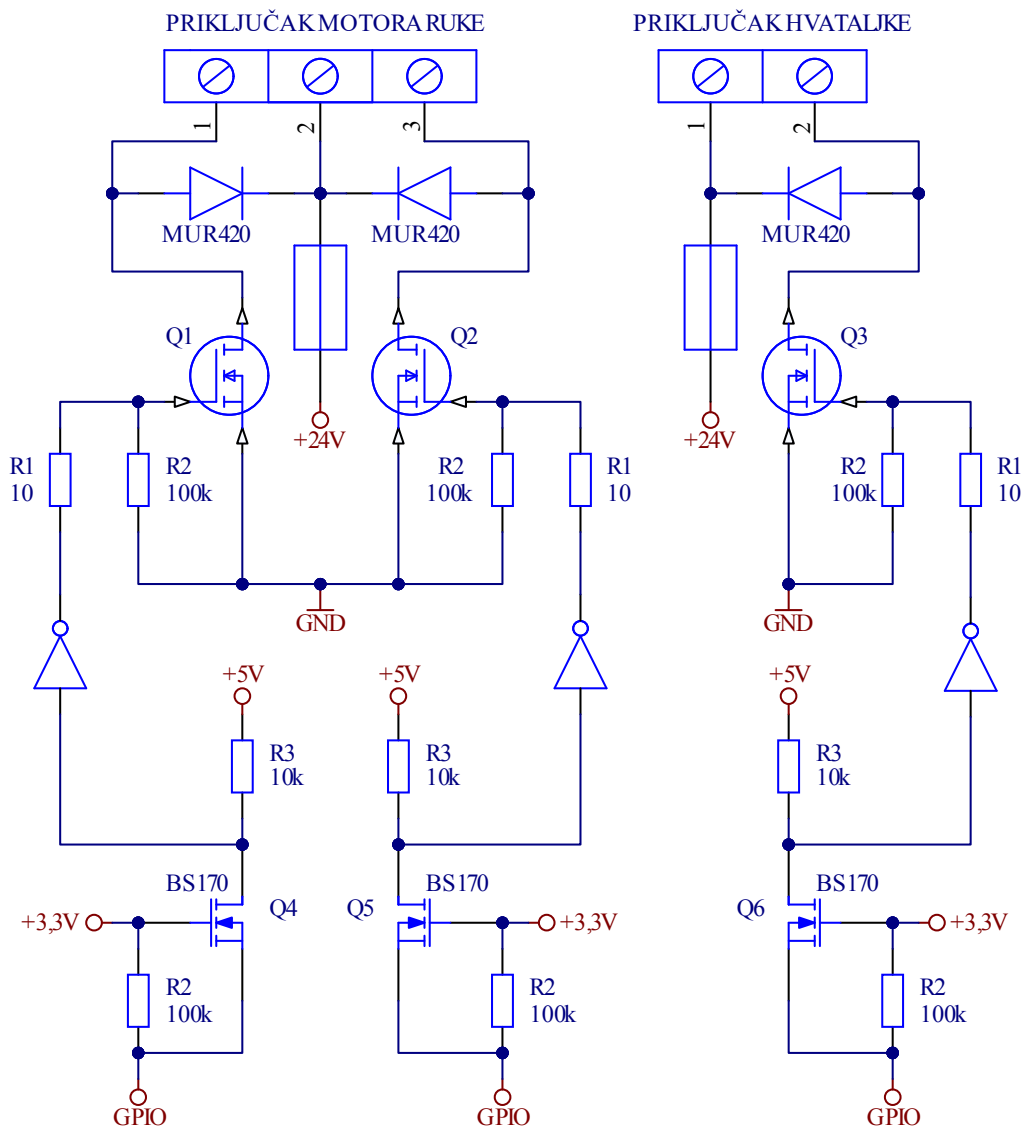
Trenutno opisana verzija upravljačkog sklopa oslanja se na pretpostavku da su stanja na izlazu GPIO priključaka fiksna i nepromjenjiva ako ih se programski ne postavi u određenu razinu. To zaista i jest tako nakon što se *Raspberry Pi* računalo pokrene, ali pitanje je u kojem su stanju GPIO priključci pri uključivanju računala. Nakon ispitivanja izlaznih napona na GPIO priključcima ustanovljeno je da se pri uključivanju računala na izlazu pojavi visoka razina na približno 3 sekunde, te se nakon toga izlazi GPIO priključaka postave u nisku razinu. Ovakva situacija predstavlja problem jer bi, kada je trošilo motor robota i hvataljka, postojao rizik da MOSFET-i na kratko provedu i dođe do kratkotrajnog, ali naglog i neželjenog trzaja ruku, glave ili hvataljki. Sljedeća stvar na koju je potrebno obratiti pažnju je brzina nabijanja kapaciteta na upravljačkoj elektrodi tranzistora IRLZ44N. Manje vrijeme nabijanja kapaciteta na upravljačkoj elektrodi omogućava precizniju kontrolu signala, smanjuje vrijeme preklapanja, a to sa sobom povlači smanjenje gubitaka preklapanja i toplinskih gubitaka, kao i manje elektromagnetske smetnje.

Kako bi se riješio problem ne definiranih stanja na izlazu GPIO pinova i brzine nabijanja kapaciteta na upravljačkoj elektrodi MOSFET-a koristi se invertirajući *buffer*. Postavljen je između upravljačke elektrode tranzistora IRLZ44N i uvida tranzistora BS170 i funkcionira na jednostavnom principu. Ako je na ulazu *buffera* visoka razina (5V), na izlazu invertora je niska razina (0V) i obrnuto. Invertirajući sklopovi su često izvedeni pomoću P-kanalnog MOSFET-a i N-kanalnog MOSFET-a kojima su obje upravljačke elektrode spojene na ulazni napon, uvid P-kanalnog MOSFET-a je spojen na napon napajanja, u ovom slučaju 5V, a uvid N-kanalnog MOSFET-a je spojen na masu. Odvodi tranzistora spojeni su u zajedničku točku koja je ujedno i izlaz invertora (Slika 3.5.). Kada je napon na ulazu 0V uključuje se P-kanalni MOSFET te je izlazni napon povezan na napon Vcc i iznosi 5V. Kada je napon na ulazu 5V uključuje se N-kanalni MOSFET te je izlazni napon povezan na masu i iznosi 0V.



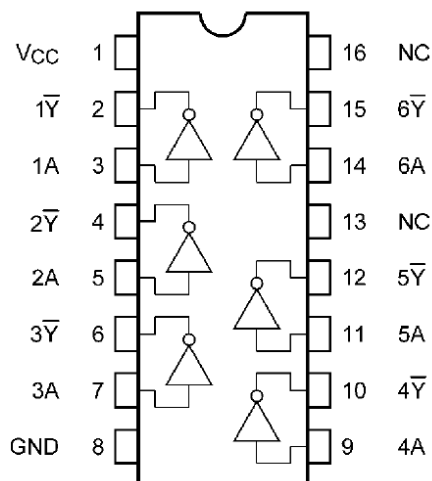
Slika 3.5. Princip rada invertora

Za upravljanje jednim motorom ruke i jednom hvataljkom potrebna su 3 invertorska sklopa, a to znači da je za cijeli sklop, koji čine dva motora ruku, dvije hvataljke i glava, potrebno 8 invertorskih sklopova koji su ujedno i naponski *bufferi*. U ovom radu je u tu svrhu korišten invertirajući naponski *buffer* CD74HC4049 [9] iz 74HC serije (*High-speed CMOS*). Ovaj integrirani krug visoke brzine ima visoku ulaznu, a nisku izlaznu impedanciju, podržava široki raspon napona napajanja (od 2V do 6V), pouzdan je i lako se integrira u upravljački sklop te sadrži šest neovisnih kanala što znači da su potrebna dva ovakva integrirana kruga za potrebe ovog rada.



Slika 3.6. Dodani invertori u shemi

Integrirani krug ima šesnaest priključaka (Slika 3.7.) koje čine šest ulaza označenih slovom A i šest izlaza označenih slovom Y, po tri sa svake strane, priključak za masu GND i za napon napajanja Vcc. Preostala dva priključka (NC) potrebno je ostaviti ne spojena. Na jedan od invertirajućih *buffera* biti će spojeni svi kanali potrebni za upravljanje motorima ruku i hvataljkama, a na drugom će biti preostala dva kanala potrebna za upravljanje motorom glave.



Slika 3.7. CD74HC4049 [9] – konfiguracija

S ovim je završen dio koji se odnosi na dizajn i ideju upravljačkog sklopa koji funkcionira na principu upravljivih sklopki, odnosno upravljivih MOSFET-a koji imaju funkcionalnost sklopke. Upravljački signali dolazit će na sklop preko GPIO priključaka *Raspberry Pi* računala.

Sklop prikazan na slici 12 testiran je na protoboard-u kako bi se uvjerali da funkcionira. Umjesto da se na pozitivan kraj napajanja dovede 24V, dovedeno je 5V, a trošila su, umjesto motora, bile LED diode kojima se upravljalo preko GPIO priključaka na *Raspberry Pi 5* računalu. Nakon što je potvrđeno da prototip sklopa funkcionira ispravno, moglo se krenuti u daljnje razvijanje upravljačkog sklopa.

3.3. Senzor položaja

Senzori položaja koji će biti postavljeni na glavu i ruke robota TIOSS bitan su dio ovog rada jer pružaju mogućnost praćenja položaja ruku i glave u stvarnom vremenu. Isto tako, omogućuje se preciznije upravljanje robotom jer je moguće zadati specifični položaj u koji ruke ili glava trebaju doći. Osim navedenih prednosti postavljanja senzora položaja također se postavljaju temelji za buduće unaprjeđenje u načinu upravljanja robotom.

U ovom radu bilo je nekoliko ideja koje su se razmatrale razmišljajući o tome kako implementirati senzor položaja na rukama i glavi. Nakon istraživanja [10] izbor se suzio na tri vrste senzora položaja: Hallov senzor, optički senzori i potencijometri.

1) *Hallov senzor položaja*

Hallovi senzori koriste magnetsko polje za detekciju položaja. Uslijed izloženosti magnetskom polju, Hallov senzor inducira napon koji je proporcionalan jačini magnetskog polja. Ovakva vrsta senzora je precizna i dugotrajna te pruža visoku pouzdanost.

2) *Optički senzor položaja*

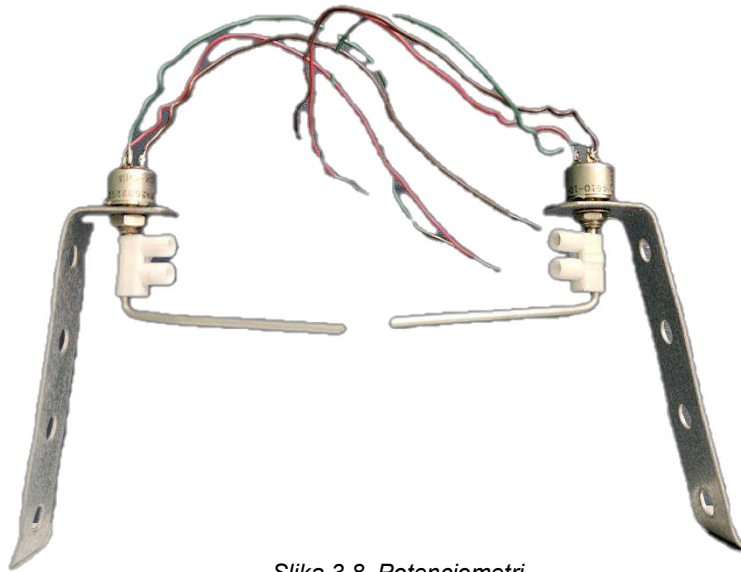
Optički senzori koriste svjetlost za detekciju položaja i sastavljeni su od izvora svjetlosti i detektora. Neki od tipova detektora svjetlosti u optičkim sensorima su fotodiode, fototranzistori i fotootpornici. Ovakvi senzori funkcioniraju na principu prekida svjetlosti kada se objekt nađe između izvora svjetlosti i detektora ili na principu promjene intenziteta svjetlosti uslijed približavanja, odnosno udaljavanja objekta od detektora svjetlosti. Prednosti ovog senzora su visoka preciznost, brza reakcija i raznolikost implementacije ovisno o primjeni.

3) *Potenciometri*

Potenciometri su promjenjivi otpornici koji imaju neku vrstu klizača kojim se mijenja otpor između dva kraja. Promjenom otpora potenciometra mijenja se i pad napona na njemu koji je lako mjerljiv, a to znači da iz izmjerene promjene pada napona doznajemo promijenjeni otpor potenciometra te time doznajemo položaj objekta koji je povezan s klizačem na potenciometru. Primjena potenciometra kao senzora položaja je jednostavna i jeftina opcija, te se može koristiti za mjerenje linearnog ili rotacijskog pomaka, a analogni izlaz se lako čita pomoću analogno-digitalnog pretvornika.

Iz navedenog, jasno je da implementacija Hallovih senzora ili optičkih senzora nije najbolje rješenje za potrebe ovog rada te da je korištenje potenciometara logičan i dobar odabir za senzore položaja ruku i glave.

Potenciometar korišten u ovom radu je ograničeni rotacijski potenciometar koji je montiran na metalno postolje te je na potenciometar stavljena mala metalna šipka koja će se kasnije montirati na ruke i glavu robota kako bi se rotacijom mijenjao otpor potenciometra (Slika 3.8.). Potenciometri se povezuju na naponski izvor od 3,3V, imaju tri priključka, jedan se spaja na pozitivni kraj napajanja, drugi se spaja na negativni kraj napajanja, a treći se spaja na analogno-digitalni pretvornik.



Slika 3.8. Potenciometri

Kako bi se očitala vrijednost koju daje potencijometar potreban je analogno-digitalni pretvornik (ADP) koji će analognu vrijednost s potencijometra pretvoriti u digitalnu. Nakon te pretvorbe potrebno je interpretirati tu vrijednost na način da se digitalna vrijednost koju daje ADP pretvori u smislenu vrijednost koja nam govori nešto o položaju ruku i glave. Ta vrijednost koju želimo promatrati je kut pod kojim se nalaze ruke, odnosno glava robota.

$$(1.) x_{ADP} \in [min_{ADP}, max_{ADP}]$$

$$(2.) x_{kut} \in [0^\circ, \alpha_{max}]$$

$$(3.) \Delta_1 = min_{ADP} - max_{ADP}$$

$$(4.) \Delta_2 = \alpha_{max} - 0^\circ$$

$$(5.) \frac{(x_{ADP} - min_{ADP})}{\Delta_1} = \frac{x_{kut}}{\Delta_2}$$

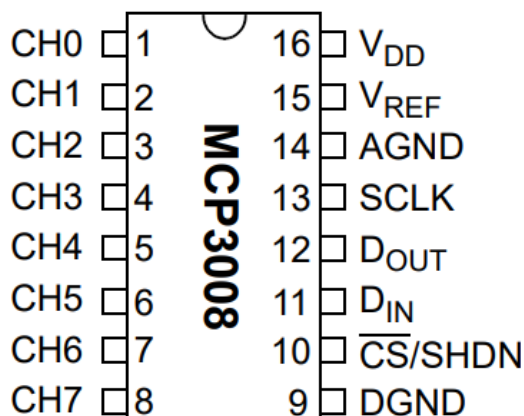
$$(6.) x_{kut} = (x_{ADP} - min_{ADP}) \frac{\Delta_2}{\Delta_1} = (x_{ADP} - min_{ADP}) \frac{\alpha_{max}}{max_{ADP} - min_{ADP}}$$

Slika 3.9. Pretvorba ADP vrijednosti u kut

Slika 3.9. prikazuje općenitu pretvorbu brojčane vrijednosti koju daje ADP iz analogne vrijednosti s potencijometra u kut, odnosno u stupnjeve. Ako nam je poznata maksimalna i minimalna vrijednost koju ADP daje (1.), znamo da će se svaka očitana vrijednost nalaziti u tom rasponu. Isto tako je potrebno znati do kojeg

se maksimalnog kuta objekt, na kojem je senzor, može kretati (2.). Sada se minimalna vrijednost ADP-a može izjednačiti s kutom 0° , maksimalna vrijednost ADP-a s maksimalnim kutom. Pomoću izjednačavanja udjela (5.) dobivamo izraz (6.) za vrijednost u stupnjevima koja nam govori pod kojim se relativnim kutom promatrani objekt nalazi.

Analogno digitalni pretvornik koji se koristio u ovom radu, odnosno upravljačkom sklopu, je MCP3008 [11] (Slika 3.10.). Isti je često korišten u projektima gdje se koriste *Raspberry Pi* računala ili *Arduino* mikrokontroleri. Dolazi u standardnom pakiranju sa šesnaest priključaka, a osam od njih su kanali ADP-a. Radi na naponima od 2,7V do 5,5V, a u ovom sklopu radi na 3,3V te mu ulazni napon može varirati od



Slika 3.10. MCP3008 – konfiguracija priključaka

0V do napona napajanja. Ovaj ADP ima 10-bitnu rezoluciju i nisku potrošnju energije. U tablici 1 prikazanoj niže pojašnjeni su priključci ovog ADP-a, a povezivanje istog s *Raspberry Pi* računalom objašnjeno je u poglavlju 3.4.

<i>CH0 - CH7</i>	Analogni ulazi za kanale ADP-a od 0 do 7
<i>V_{DD}</i>	Napon napajanja
<i>V_{REF}</i>	Referentni napon za određivanje raspona analognog napona
<i>AGND</i>	Analogno uzemljenje
<i>SCLK</i>	Serial clock odgovoran za sinkronizaciju podataka
<i>D_{OUT}</i>	Izlaz serijskih podataka za SPI komunikaciju
<i>D_{IN}</i>	Ulaz za serijske podatke
<i>$\overline{CS}/SHDN$</i>	Priključak za odabir čipa
<i>DGND</i>	Digitalno uzemljenje

Tablica 3.1. Priključci

3.4. Mehaničke sklopke za ograničavanje kretnje ruku i glave

Osim što su se potenciometri postavili na robota, također su postavljene mehaničke sklopke, odnosno graničnici koji osiguravaju da, kada ruke ili glava dođu do krajnjeg položaja, ne dođe do oštećenja konstrukcije robota ili samih motora s obzirom da je raspon kretnji ruku i glave fizički ograničen (Slika 3.11.). Parovi sklopki postavljeni su na svaku ruku i glavu robota kako bi oba krajnja položaja bila osigurana.



Slika 3.11. Postavljena mehanička sklopka na rame robota

Kada ruka dođe do krajnjeg položaja, pritisne nožicu sklopke i time se strujni krug prekine tako da se motori više ne rotiraju.

3.5. *Raspberry Pi* računalo

Raspberry Pi je popularno, malo i pristupačno računalo te pruža veliki broj funkcionalnosti za različite primjene poput promocije računalnog obrazovanja, različitih pokaznih projekata ili specifičnijih primjena kao što je ovaj rad. Ova računala razvija *Raspberry Pi Foundation*, a njihovo prvo komercijalno dostupno računalo *Raspberry Pi Model B* stiglo je na tržište u veljači 2012. godine [12].

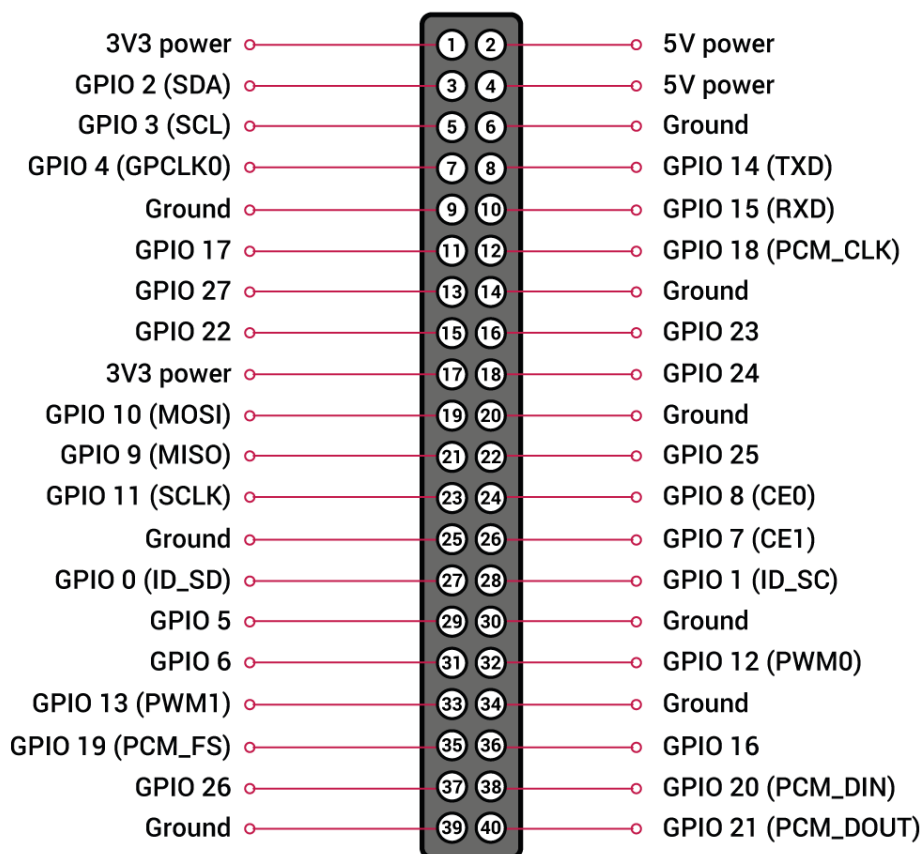
U robotu TIOSS se, još prije ovog rada, nalazila ploča s upravljačkim sklopom kojim se upravljalo *Raspberry Pi 4* računalom, a tijekom ovog rada korištene su dvije inačice *Raspberry Pi* računala: *Raspberry Pi 3* i *Raspberry Pi 5*. Obje inačice bile su dovoljno dobre te su pružale sve potrebne funkcionalnosti za potrebe ovog projekta, a Tablica 3.2. prikazuje usporedbu ključnih značajki obje inačice.

Ovdje je važno naglasiti jednu bitnu razliku između inačica *Raspberry Pi* računala koja je u početku dovela do poteškoća u pisanju programskog koda za upravljanje sklopom, odnosno robotom TIOSS. Naime, *Raspberry Pi 5* je uveo značajne hardverske promjene te sada koristi novi, drugačiji čip za GPIO periferne jedinice [7]. Posljedica koja proizlazi iz korištenja drugačijeg čipa za GPIO je različito mapiranje GPIO perifernih jedinica. Programski dio ovoga rada detaljnije je opisan u poglavlju 5., ali ovdje je bitno napomenuti da zbog različitog načina mapiranja GPIO perifernih jedinica neke od biblioteka, među kojima su i one koje su se najučestalije koristile na prethodnim inačicama, više nisu kompatibilne sa *Raspberry Pi 5* inačicom te se ne mogu koristiti.

Iako se najnovija inačica *Raspberry Pi* računala razlikuje po načinu mapiranja GPIO perifernih jedinica, 40 priključaka ostali su isti kao i na prethodnim inačicama računala (Slika 3.12.). Kako raspored, tako i funkcionalnost priključaka ostao je nepromijenjen kroz inačice računala, a to je značajno olakšalo korištenje različitih inačica tijekom razvoja rješenja s obzirom da se jedino trebalo prilagoditi programski dio, odnosno, bilo je potrebno promijeniti korištene biblioteke u programskom kodu s obzirom da se jednaka funkcionalnost može postići, neovisno o korištenim bibliotekama i korištenoj inačici računala.

	<i>Raspberry Pi 3</i>	<i>Raspberry Pi 5</i>
<i>Procesor</i>	Quad-core 64-bit ARM Cortex-A53	Quad-core 64-bit ARM Cortex-A53
<i>RAM</i>	1 GB LPDDR2 Takt: 1.2 GHz	4 GB ili 8 GB LPDDR4 Takt: 1.8 GHz
<i>Grafička kartica</i>	Broadcom VideoCore IV	Broadcom VideoCore VII
<i>Pohrana</i>	MicroSD kartica	MicroSD kartica
<i>USB priključci</i>	Četiri USB 2.0 priključka	Dva USB 3.0 Dva USB 2.0
<i>Mreža</i>	- <i>Wi-Fi</i> : 802.11n (2.4GHz) - <i>Bluetooth</i> : 4.1 - <i>Ethernet</i> : 10/100 Mbps	- <i>Wi-Fi</i> : 802.11ac (dual-band 2.4GHz i 5GHz) - <i>Bluetooth</i> : 5.0 - <i>Ethernet</i> : Gigabit Ethernet
<i>Video izlaz</i>	HDMI i Video kompozitni kabel (3.5mm)	2 x micro-HDMI (podržava 4K na 60Hz)
<i>Operacijski sustav (OS)</i>	Primarno Raspbian, ali i drugi OS prilagođeni za ARM arhitekturu	Raspbian, Ubuntu i drugi OS prilagođeni za ARM arhitekturu
<i>Ulazno/Izlazni priključci</i>	40 priključaka	40 priključaka

Tablica 3.2. Usporedba inačica Raspberry Pi računala [7]



Slika 3.12. Ulazno/Izlazni priključci Raspberry Pi računala

Raspberry Pi računala imaju 40 ulazno/izlaznih priključaka koji omogućuju različite funkcionalnosti [7]. Osam je priključaka za masu (*Ground*) dok četiri priključka pružaju izvor napajanja. Dva na izlazu daju 3,3V (*3V3 power*), a druga dva daju 5V (*5V power*). Važno je biti pažljiv pri korištenju izvora napajanja i masa kako ne bi došlo do kratkog spoja koji nebi trebao naštetiti računalu, ali će zasigurno uzrokovati isključivanje računala. Dva priključka se ističu s obzirom na specifičnu funkcionalnost, a to su ID EEPROM priključci *GPIO 0 (ID_SD)* i *GPIO 1 (ID_SC)*. Naime, ova dva priključka nisu namijenjena za uobičajenu upotrebu te su povezana s prepoznavanjem HAT (*Hardware Attached on Top*) uređaja korištenjem EEPROM memorije (*Electrically Erasable Programmable Read-Only Memory*). To su uređaji koji se dodatno povezuju s *Raspberry Pi* računalom kako bi se proširila funkcionalnost samog računala. *GPIO 0 (ID_SD)* koristi se za podatkovnu liniju I2C (*Inter-Integrated Circuit*) sabirnice koja čita ID EEPROM-a na HAT uređajima, a *GPIO 1 (ID_SC)* se koristi za liniju sata I2C sabirnice. Ovi priključci se najčešće ostavljaju neiskorišteni, ako nisu potrebni, kako ne bi došlo do problema kod prepoznavanja HAT uređaja.

Preostalih 26 GPIO (eng. *General Purpose Input/Output*) priključaka su višenamjenski, ali neki od njih imaju predviđenu namjenu dok su drugi potpuno slobodni za bilo kakvu primjenu. Priključci koji imaju predviđenu namjenu (Tablica 3) mogu se prenamijeniti i koristiti u druge svrhe. Kao što im ime govori, GPIO priključci mogu se koristiti kao izlazni ili ulazni priključci, a kada se koriste kao izlazni razlikujemo dvije razine:

- Niska razina (0) = 0V
- Visoka razina (1) \approx 3,3V

Raspon napona na kojem GPIO priključci funkcioniraju sigurno je od 0V do 3,3V. Kada se priključci koriste kao ulazni postoje naponi praga za koje su definirana logička stanja koja priključci prepoznaju:

- Niska razina (0) \in [0V, 0,8V]
- Visoka razina (1) \in [1,3V, 3,3V]

<i>Priključak</i>	<i>Funkcionalnost</i>
<i>GPIO 2 (SDA)</i>	Serijska podatkovna linija korištena za serijski prijenos podataka
<i>GPIO 3 (SCL)</i>	Serijska linija sata korištena za sinkronizaciju prijenosa podataka
<i>GPIO 4 (GPCLK0)</i>	(General Purpose Clock Output) generički takti signal
<i>GPIO 10 (MOSI)</i>	„Master Out Slave In“ – slanje podataka s računala na periferije
<i>GPIO 9 (MISO)</i>	„Master In Slave Out“ – primanje podataka s periferija na računalo
<i>GPIO 11 (SCLK)</i>	Serijska linija sata korištena za sinkronizaciju prijenosa podataka između „master“ i „slave“ uređaja
<i>GPIO 13 (PWM1)</i>	Generiranje PWM signala za upravljanje analognim uređajima
<i>GPIO 19 (PCM_FS)</i>	„Frame Sync“ – koristi se za sinkronizaciju početka svakog okvira podataka
<i>GPIO 14 (TXD)</i>	„Transmit data“ - Slanje podataka s <i>Raspberry Pi</i> računala na drugi uređaj
<i>GPIO 15 (RXD)</i>	„Recieve data“ - Prijem podataka na <i>Raspberry Pi</i> računalo s drugog uređaja
<i>GPIO 18 (PCM_CLK)</i>	„Clock“ – koristi se za sinkronizaciju prijenosa podataka
<i>GPIO 8 (CE0)</i>	Odabir prvog „slave“ uređaja na SPI sabirnici
<i>GPIO 7 (CE1)</i>	Odabir drugog „slave“ uređaja na SPI sabirnici
<i>GPIO 12 (PWM0)</i>	Generiranje PWM signala za upravljanje analognim uređajima
<i>GPIO 20 (PCM_DIN)</i>	„Data In“ – koristi se za primanje PCM podataka na računalo
<i>GPIO 21 (PCM_DOUT)</i>	„Data Out“ – koristi se za slanje PCM podataka sa računala

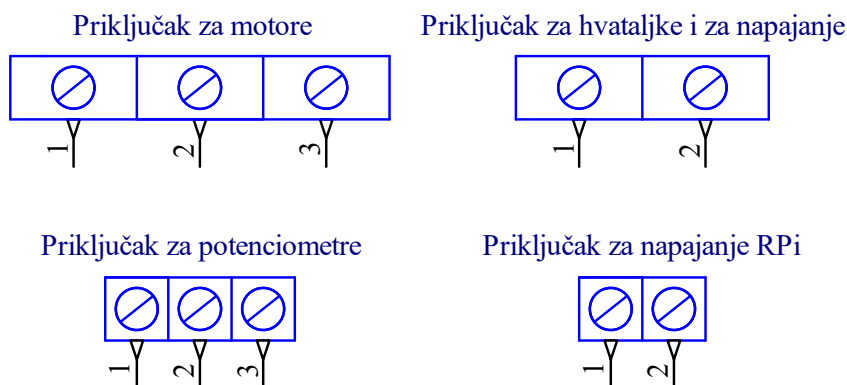
Tablica 3.3. Specifične namjene pojedinih GPIO priključaka [7]

Napomene:

- PCM (*Pulse Code Modulation*) je metoda pretvorbe analognih signala u digitalne, često korištena za prijenos audio podataka
- Prijenos podataka putem TXD i RXD realizira se UART protokolom
- PWM (*Pulse Width Modulation*) je metoda koja se koristi za upravljanje analognim uređajima te omogućuje varijabilno upravljanje snagom koja se daje tim uređajima
- SPI (*Serial Peripheral Interface*) predstavlja serijsku komunikaciju s periferijama

3.6. Priključci na upravljačkom sklopu

Na upravljačkom sklopu u ovom radu korišteni su različiti priključci za povezivanje motora, hvataljki, napajanja, pretvarača napona (opisan u poglavlju 4.1.) *Raspberry Pi* računala i senzora položaja, odnosno potencijometara. Slika 3.13. prikazuje shematske prikaze priključaka, a Slika 3.14. prikazuje stvarne priključke koji su korišteni za povezivanje potencijometara s upravljačkim sklopom.

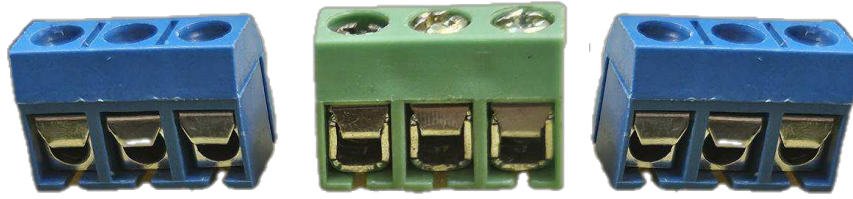


Slika 3.13. Shematski prikaz priključaka

S obzirom da se na priključak za napajanje dovodi napon iznosa 24V, kao i na priključak za motore i hvataljke, razmak između pojedinih priključaka za žice je nešto veći nego kod priključaka za potencijometre, pretvarač napona i *Raspberry Pi* računalo na koje se dovodi napon iznosa 5V. Razmak između pojedinih priključaka na priključku za motore, hvataljke i napajanje je približno 10mm, a na priključku za potencijometre, pretvarač napona i *Raspberry Pi* je približno 5mm.

Priključci s razmakom pojedinačnih priključaka od 5mm su tako i proizvedeni, a korištene su varijante s dva i tri pojedinačna priključka. Varijanta s dva pojedinačna priključka korištena je za pretvarač napona i za napajanje *Raspberry Pi* računala, dok je varijanta s tri pojedinačna priključka korištena za povezivanje potencijometara s obzirom da potencijometri imaju pozitivan i negativan kraj napajanje, ali i treći priključak koji vodi na jedan od kanala ADP-a.

Za potrebe ovog rada, priključci s razmakom pojedinačnih priključaka od 10mm su napravljeni iz jednog većeg priključka u varijante s dva ili tri pojedinačna priključka. Bilo je potrebno samo izrezati veći priključak na nekoliko manjih i zagladiti rubove.



Slika 3.14. Priključci za potenciometre

4. Izrada Pločice za upravljački sklop

Nakon što je ideja za upravljački sklop razrađena, potrebno je bilo napraviti konačnu električnu shemu na kojoj će sve biti objedinjeno te potom krenuti s dizajnom PCB (eng *Printed Circuit Board*) pločice. Za stvaranje dizajna PCB pločice potreban je program za dizajn, a u svrhu izrade pločice za upravljački sklop kojim će se upravljati rukama i glavom robota TIOSS korišten je *Altium Designer (Version 22.0.2)* [13].

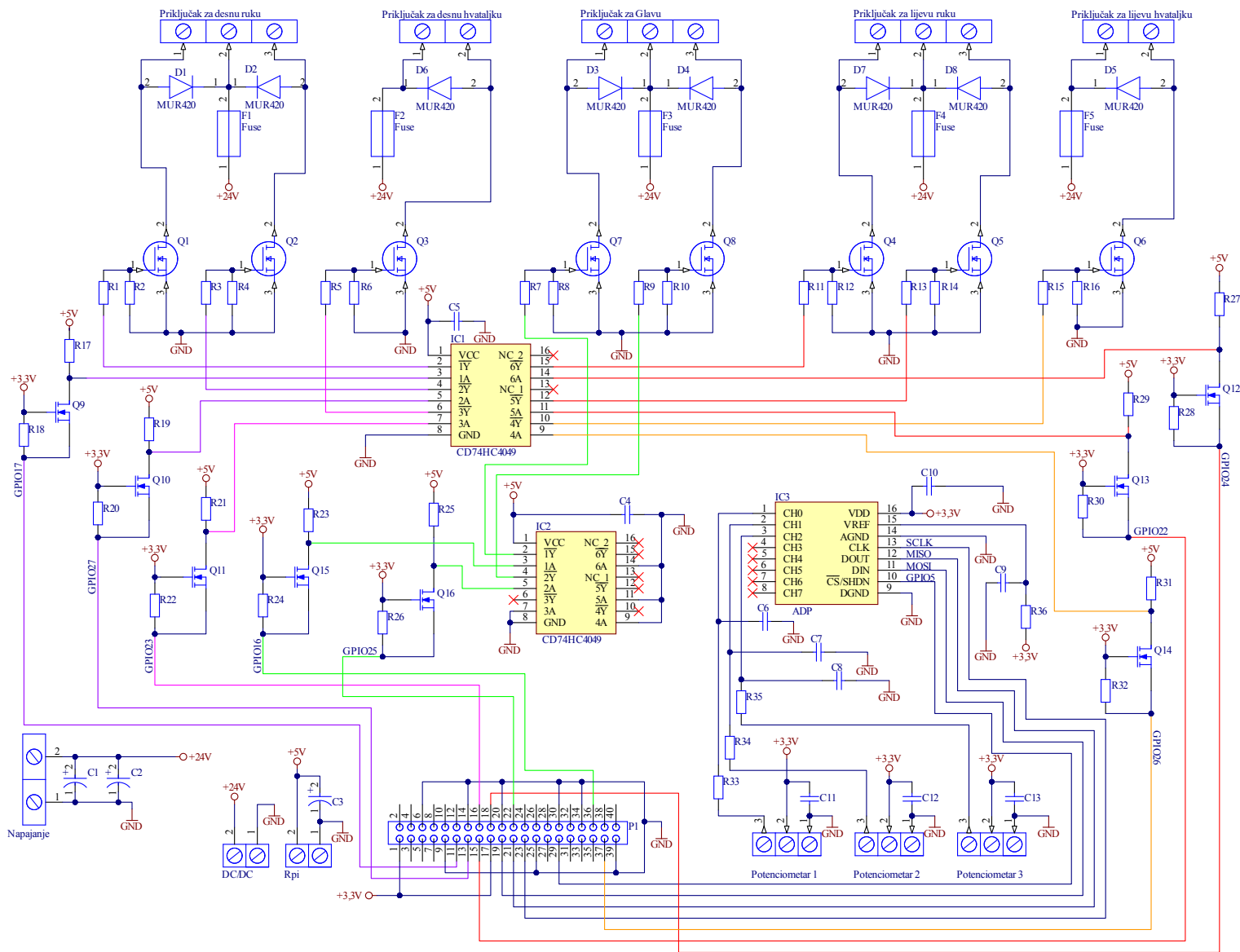
Altium Designer nije jedini program za dizajn PCB pločica, ali je nedvojbeno jedan od vodećih. Osim što se koristi za dizajn PCB-a, koristi se i za dizajn raznih elektroničkih sustava, a razvio ga je *Altium Limited*, američko-australska kompanija koja je osnovana u Australiji 1985. godine, danas s regionalnim središtima u SAD-u, Australiji, Japanu, Kini i Europi [14].

4.1. Kompletna električna shema

U poglavlju 3.2 opisan je upravljački sklop za motor ruke i hvataljku, a time i za motor glave. Objedinjena shema (Slika 4.1.) sadrži upravljačke sklopove za oba motora ruku, obje hvataljke i za motor glave. Također su postavljeni invertirajući naponski *bufferi* CD74HC4049 kao i ADP pretvornik MCP3008. Na shemi su na samome vrhu postavljeni priključci za povezivanje motora ruku i glave te za povezivanje hvataljki, a pri dnu su postavljeni priključci za senzore položaja, odnosno potenciometre, priključci za napajanje, pretvarač napona i napajanje *Raspberry Pi* računala i konačno komponenta za povezivanje 40 ulazno/izlaznih priključaka *Raspberry Pi* računala. Radi lakšeg raspoznavanja, žice koje povezuju komponente na shemi su različitih boja te na pojedinim žicama stoji naglašeno na koji priključak računala

vode. Važno je obratiti pažnju na nekoliko dodatnih komponenti u električnoj shemi, a to su:

- RC članovi za spajanje potencimetara na kanale ADP-a
- Blokadni kondenzatori na priključcima za napajanje invertirajućih naponskih *buffera* i ADP-a te između 3,3V i mase kod priključaka za potencimetre
- Elektrolitski kondenzatori postavljeni između 24V i mase te između 5V i mase

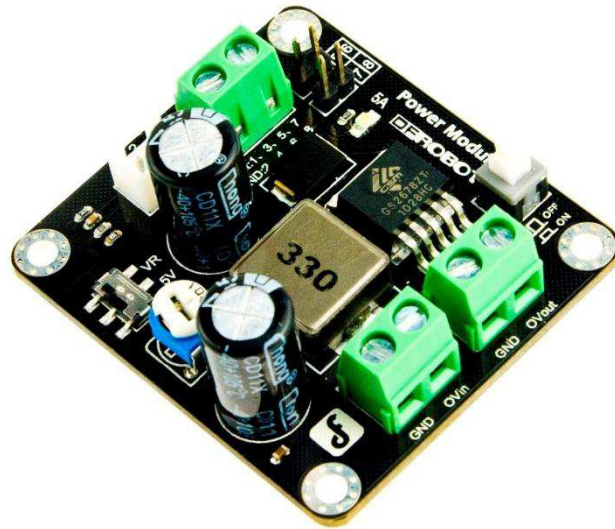


Slika 4.1. Kompletna električna shema upravljačkog sklopa

Između priključka za povezivanje potenciometara i kanala ADP-a postavljeni su RC članovi koji se za ovaj slučaj koriste u svrhu uklanjanja šuma. S obzirom da je ulaz na kanal ADP-a preko kondenzatora, radi se o niskopropusnom filtru za uklanjanje visokofrekvencijskog električnog šuma. Otpornici tih RC članova iznose $470\text{k}\Omega$, a kondenzatori su kapaciteta 10nF što bi značilo da vremenska konstanta τ iznosi $4,7\text{ms}$. Između priključaka za napajanje sklopova CD74HC4049 i MCP3008 kao i između napajanja $3,3\text{V}$ i mase za priključke potenciometara postavljeni su kondenzatori kapaciteta 100nF . Ti kondenzatori nazivaju se blokadnim kondenzatorima (eng. *Bypass Capacitors*) i tipično služe za filtriranje visokofrekvencijskog šuma koji dolazi s linije napajanja, stabilizaciju napajanja i preusmjeravaju smetnje na masu kako bi se spriječilo njihovo širenje kroz ostatak sklopa. Posljednji tip kondenzatora na ovom sklopu su elektrolitski kondenzatori kapaciteta $1000\mu\text{F}$. Takav tip kondenzatora se tipično koristi za stabilizaciju napajanja pri naglim promjenama opterećenja, ali i za privremeno skladištenje energije kako bi pružili brz izvor energije kada je to potrebno.

Dosad se kroz više poglavlja spominjao pretvarač napona, a na konačnoj shemi postavljeni su i priključci *DC/DC* i *Rpi*. Priključci su postavljeni zbog istosmjernog pretvarača napona korištenog u upravljačkom sklopu koji služi za pretvaranje napona od 24V , kojeg mu se dovodi na ulaz, u napon od 5V koji je na izlazu pretvarača. Napon sa izlaza pretvarača koristi se za napajanja *Raspberry Pi* računala, ali se taj napon od 5V dovodi i na ostale ključne točke na upravljačkom sklopu.

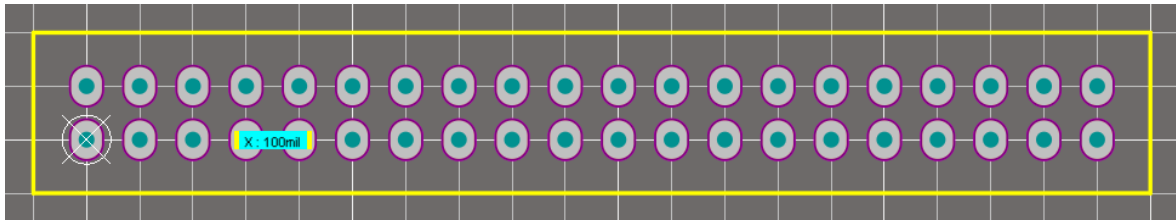
Pretvarač napona koji je korišten u ovom radu je DFR0205 [15], proizveden od strane DFRobot-a (Slika 4.2.). Radi se o istosmjernom pretvaraču napona kojem je raspon ulaznog napona od $3,6\text{V}$ do 25V , a izlazni napon je podesiv i može iznositi od $3,3\text{V}$ do 25V . Izlazni napon se pomoću sklopke može fiksirati na 5V ili ga se može podešavati na željeni napon pomoću rotacijskog potenciometra. Maksimalna izlazna snaga pretvarača je 24W . Za potrebe upravljačkog sklopa u ovom radu napon koji se dovodi na ulaz pretvarača je napon kojega daje serijski spoj dvaju akumulatora (12V svaki) te iznosi 24V , a na izlazu pretvarača je napon iznosa 5V .



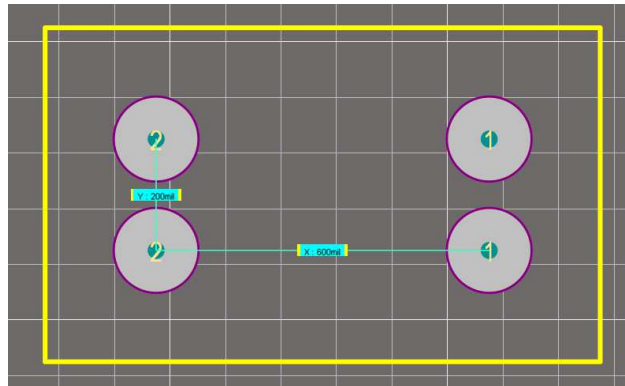
Slika 4.2. DC/DC pretvarač napona DFR0205

4.2. Priprema za PCB dizajn

Prije no što se krene s projektiranjem PCB dizajna potrebo je pobrinuti se da svaka komponenta sa električne sheme ima svoj isparavan *footprint*. Naime, *footprint* je predložak prema kojem se u procesu fizičke izrade PCB pločice buše rupe, postavljaju i električki povezuju komponente. Da bi sve u konačnici bilo dobro električki povezano, važno je da je veličina rupe za priključke pojedine komponente manja od veličine *padova* na *footprintu* kako bi nakon bušenja rupe ostalo dovoljno okolnog bakra za povezivanje i lemljenje komponente na pločicu. Svakoj komponenti na električnoj shemi može se pridijeliti njen *footprint*, a neke od komponenata, poput CD74HC4049 invertirajućega naponskog *buffera* ili ADP-a MCP3008, već imaju vlastiti *footprint*. Za druge komponente sa sheme, posebno priključke, bilo je potrebno nacrtati njihove *footprinte* i pridružiti ih komponentama u shemi. Kao što je već spomenuto, razmak između *padova* (rupa na *padovima*) kod priključaka za motore, hvataljke i napajanje iznosi 400mil ($\approx 10\text{mm}$), a kod priključaka za potencioetre, pretvarač napona i napajanje za *Raspberry Pi* razmak iznosi 200mil ($\approx 5\text{mm}$). Također je bilo potrebno nacrtati *footprint* za 40 ulazno/izlaznih priključaka (Slika 4.3.) i za držače osigurača (Slika 4.4.).



Slika 4.3. Footprint za 40 ulazno/izlaznih priključaka



Slika 4.4. Footprint za držače osigurača

4.3. PCB dizajn

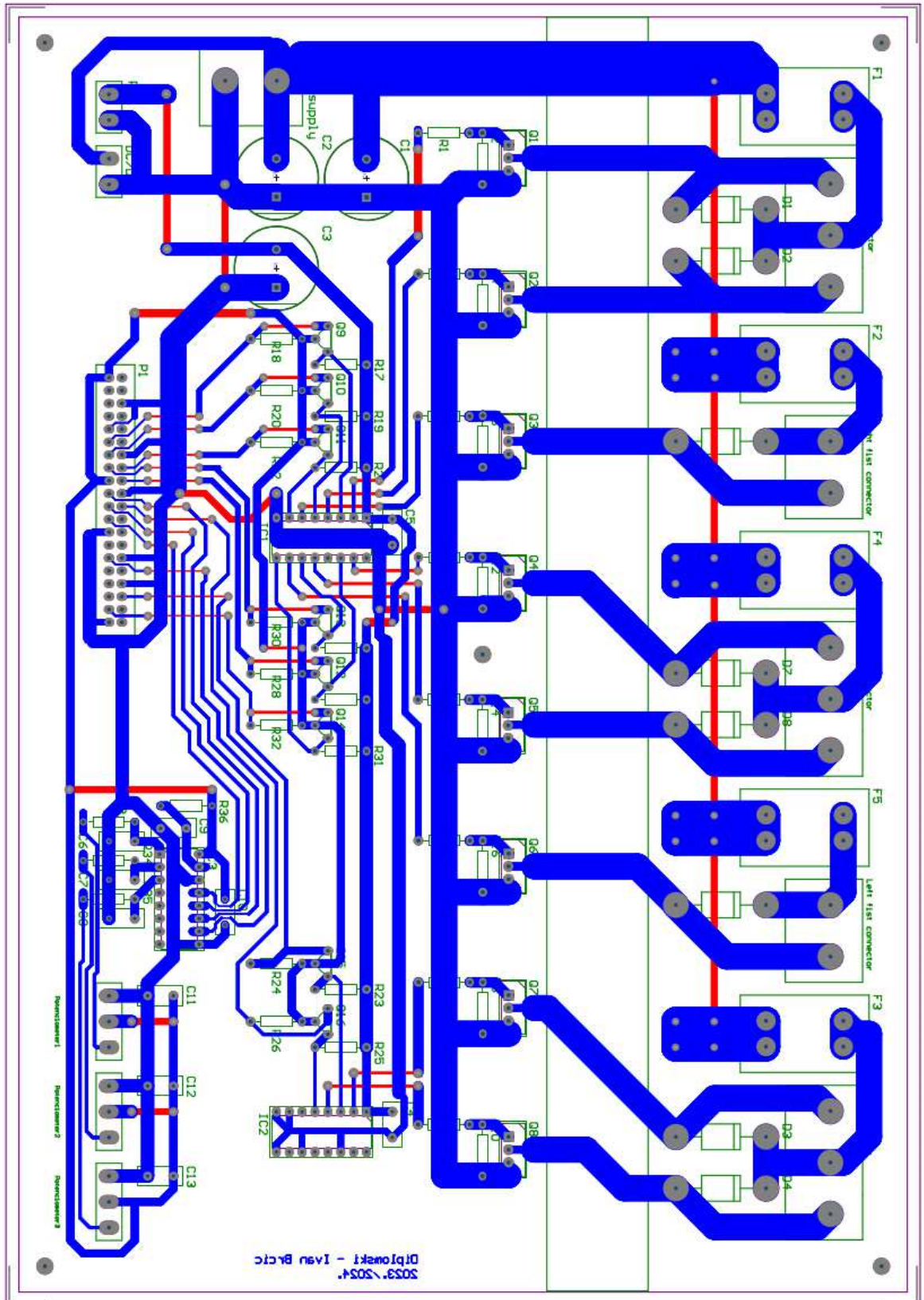
Kada je kompletna električna shema dovršena i kada svaka komponenta sa sheme ima svoj isparavan *footprint*, može se započeti s PCB dizajnom.

Za početak, u *Altium Designeru* je potrebno stvoriti novi *.PcbDoc* i definirati dimenzije pločice i *Keep-Out Layer* koji predstavlja dio na kojem se nikakva komponenta, rupa ili vod ne smije pojaviti. Za opisani upravljački sklop širina pločice iznosi 10100mil ($\approx 256\text{mm}$), a visina 7100mil ($\approx 180\text{mm}$). Na prvi pogled, dimenzije pločice su poprilično velike, ali vodovi koji će se provlačiti pločicom biti će također poprilično široki tako da kada bi htjeli smanjiti pločicu, ne bi bilo dovoljno mjesta za provlačenje širokih vodova. Naravno, PCB pločice mogu se raditi u više slojeva, a to olakšava povezivanje komponenti kroz slojeve i daje mogućnost da sama pločica bude manjih dimenzija. Za potrebe ovog rada odlučeno je da će se PCB pločica raditi na prostorijama fakulteta i da će pločica biti jednoslojna, te je upravo iz tih, praktičnih razloga, pločica dobila takve dimenzije. *Keep-Out Layer* je odmaknut od rubova pločica za 100mil (2,54mm), a uz kutove *Keep-Out Layer-a*, ali i u samo središte pločice, postavljeni su *padovi* za rupe kojima će se pločica pričvrstiti za

drugu ploču na kojoj će uz PCB pločicu biti postavljeni i pretvarač napona te *Raspberry Pi* računalo. Na shemi je također naznačeno područje koje je ostavljeno prazno, kako bi se u budućnosti, po potrebi, moglo postaviti hladilo za tranzistore. Nakon toga, potrebno je prenijeti električnu shemu (.SchDoc) u PCB shemu slijedeći ove korake: **Design → Import Changes From ____PrjPcb**

Nakon toga, sve komponente, odnosno svi footprinti komponenata trebali bi biti preneseni u PCB shemu te bi žutim linijama trebalo biti naznačeno koju komponentu treba spojiti na koju točku.

Kada su svi *footprinti* preneseni na shemu slijedilo je povezivanje komponenata vodovima. Potrebno je bilo pozicionirati se u *Bottom Layer*, označen plavom bojom, i krenuti s povezivanjem komponenata. Važno je bilo osigurati da su vodovi za masu i za napajanje, posebno ono iznosa 24V, dovoljno široki. Na nekim mjestima je bilo moguće postaviti ih da budu širi, a na drugim mjestima ne, ali nisu bili uži od 100mil. S obzirom da se pločica izrađuje u jednom sloju, u situacijama gdje nije bilo jednostavno povezati određene točke na PCB shemi, korišteni su kratkospojnici koji su na shemi (Slika 4.5.) prikazani crvenom bojom, označavajući da se nalaze na *Top Layer-u*. Proces razmještanja i povezivanja komponenata provlačenjem vodova bio je pun izmjena jer je uvijek moguće na malo bolji način napraviti razmještaj komponenata na pločici. Važan detalj kod razmještanja komponenti na shemi koji je potrebno naglasiti je položaj komponenata vezanih za senzore položaja i ADP. S obzirom da se na ADP dovode analogni signali, ideja je bila donekle odvojiti cijeli dio povezan sa sensorima položaja u svoj kut. To je i napravljeno te se analogno-digitalni pretvornik MCP3008, priključci za potencioetre i RC članovi nalaze u donjem desnom kutu pločice kao mala zasebna cjelina. Vodovi kojima se povezivalo komponente s GPIO priključcima nisu široki jer su struje koje njima idu poprilično male, ali su zato svi priključci mase sa *Raspberry Pi* računala zajednički povezani.



Slika 4.5. PCB shema upravljačkog sklopa

4.4. Izrada PCB pločice

Postupak izrade PCB pločice odvijao se na Fakultetu elektrotehnike i računarstva, na Zavodu za elektroničke sustave i obradbu informacija (ZESOI) u mehaničkoj radionici D148 prema uputama za izradu tiskane pločice [16]

Za izradu pločice korišten je A4 komad vitroplasta (materijal sastavljen od staklenih vlakana i epoksi smole) koji je potrebno dobro prebrisati ručnikom i neprekinutim mlazom, jednoliko i u tankom sloju nanijeti prethodno dobro protreseni lak. Potom, pločica se stavlja na sušenje u pećnicu na temperaturu oko 70 °C nekoliko minuta. Potom na komad vitroplasta postaviti prozirni papir na kojem je isprintan *Bottom Layer* PCB sheme te ga, filmom prema dolje postaviti u uređaj za osvjetljivanje i ostaviti ga unutar uređaja na otprilike 40 sekundi. U malu kadicu ulivena je lužina (natrijev hidroksid) i dodano je malo vode kako nebi pretjerano agresivno skinula sloj laka s pločice koja se potopi u lužinu. Postepeno se s pločice dio laka koji je bio osvijetljen, a onaj koji nije, odnosno oslikani *Bottom Layer*, ostaje na pločici. Nakon toga, pločica je isprana vodom i posušena (Slika 4.6.), a sljedeći korak bio je proces jetkanja. To je proces skidanje bakra s pločice, ali samo s onog dijela na kojem se lak maknuo nakon uranjanja u feriklorid. Pločica je pričvršćena na držač u stroju za jetkanje, poklopac je spušten i stroj se mogao pokrenuti. Stroj stvara pjenu koja prelazi preko pločice te na taj način, kemijskom reakcijom ($Cu + 2FeCl_3 = CuCl_2 + 2FeCl_2 \rightarrow Cu + CuCl_2 = 2CuCl \rightarrow CuCl + FeCl_3 = CuCl_2 + FeCl_2$) skida bakar s pločice. Po završetku jetkanja, pločica je ponovno isprana vodom i posušena ručnikom (Slika 4.7.).



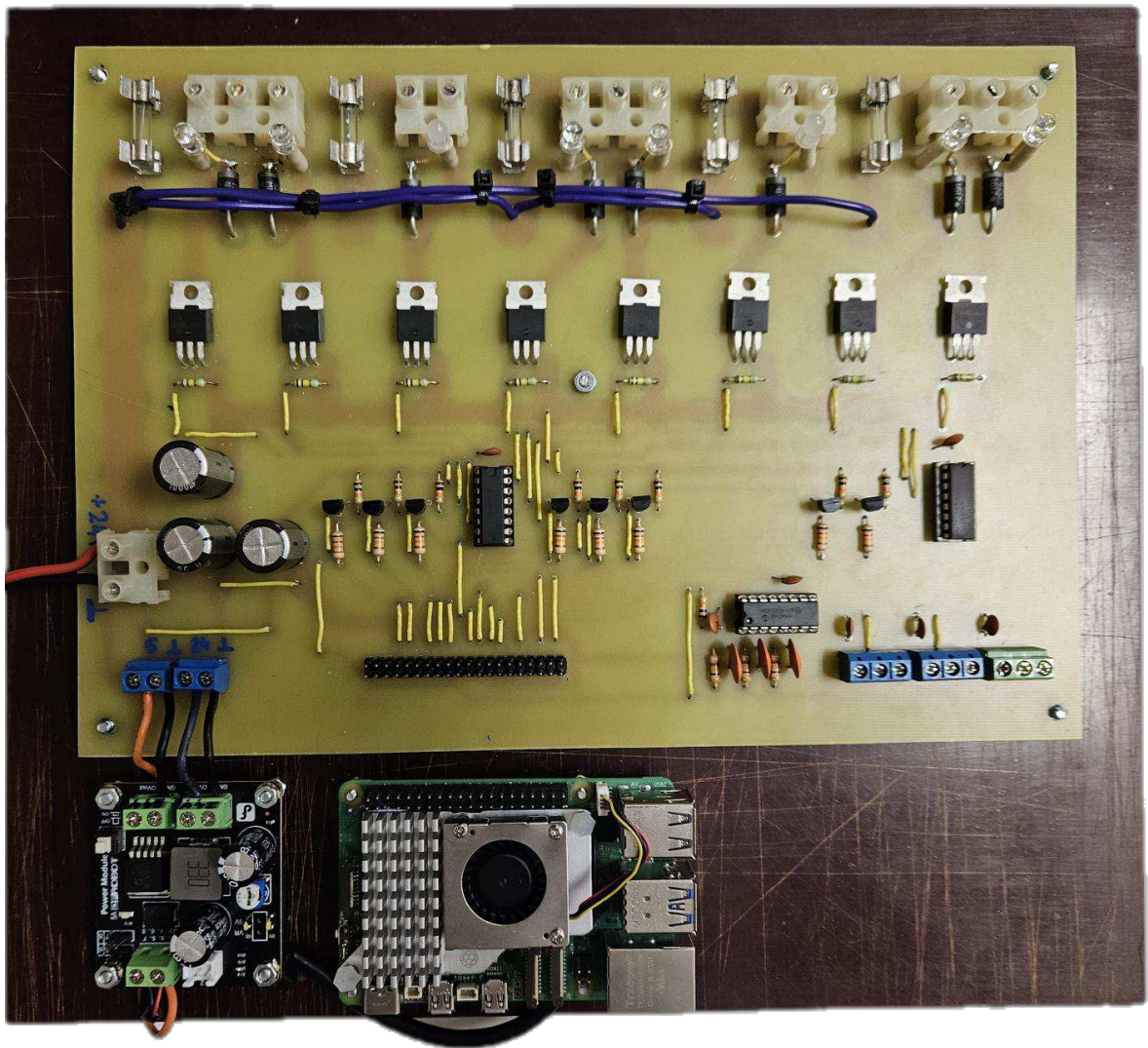
Slika 4.6. PCB prije jetkanja



Slika 4.7. PCB nakon jetkanja

Sav nepotreban bakar s pločice je maknut, a dijelovi bakra koji su ostali na pločici oslikavaju *Bottom Layer* PCB sheme. Pločica je izrezana na dimenzije navedne u opisu PCB sheme prema ocrtanim rubovima, a onda je uslijedilo bušenje rupa na mjestima gdje bili postavljeni *footprinti* komponenti. Inače, radi olakšavanja procesa bušenja rupa, običaje je napraviti plan bušenja kako bi se taj dio izrade pločice odradio što bolje i sistematiziranije. Rupe su izbušene u radioni koristeći svrdla različitih debljina, za poneke komponente promjer rupe iznosio je 0,8mm dok je za druge promjer rupe iznosio 1mm. Promjer rupa za pričvršćivanje pločice na veću ploču nešto je veći te iznosi 3mm. Nakon što su rupe izbušene na pravim mjestima, nastupilo je lemljenje komponenti na donjoj strani pločici s obzirom da su sve komponente *through-hole* komponente. Običaj je da se komponente se na pločicu leme od najnižih prema najvišim, kako bi proces lemljenja bi što jednostavniji i praktičniji. Pri lemljenju komponenti došlo je do jedne male izmjene. Naime, umjesto otpornika, planirano otpora 10Ω , koji su se na električnoj i PCB shemi nalazili između izlaza iz invertirajućih naponskih *buffera* i upravljačke elektrode tranzistora IRLZ44N zamijenjeni su kratkospojnicima jer je procijenjeno da nije nužno da na tom mjestu budu otpornici ionako malenog iznosa. S donje strane pločice je na glavnim, najširim vodovima za masu zalemljena dodatna žica kako bi ti vodovi bili još sigurniji i stabilniji. Također, dodano je nekoliko komponenti, točnije njih pet gotovo identičnih, a to su LED diode različitih boja i njima u seriju spojeni veliki otpori. Dioda su postavljene uz priključke za motore ruku i glave te uz priključke za hvataljke, na mjesta između pozitivnog kraja napajanja i vodova koji vode prema masi preko N-kanalnih MOSFET-a. Postavljene su kako bi testiranje ispravnosti sklopa bilo lakše, a zapravo simuliraju i ukazuju dolazi li napon ispravno na priključke, odnosno hoće li napon ispravno dolaziti na motore i hvataljke. Isto tako, ideja je bila da LED diode pokazuju koji dio sklopa je uključen a koji dijelovi nisu čak i onda kada se cijela ploča s upravljačkim sklopom postavi u robota.

S obzirom da je PCB pločica bila završena, na drugu, veću ploču probušene su rupe na mjestima gdje su se onda postavili i pričvrstili istosmjerni pretvarač napona, *Raspberry Pi* i PCB pločica kako bi se sve nalazilo na istom mjestu i bilo fiksirano i spremno za postavljanje u robota.



Slika 4.8. Završen i povezan upravljački sklop

5. Programska podrška za upravljanje robotom

Raspberry Pi računala omogućuje programiranje u više različitih programskih jezika, a neki od njih su *Python*, *Scratch*, *Java*, *C/C++* i drugi. Za upravljanje robotom TIOSS putem upravljačkog sklopa opisanog u ovom radu korišten je *Python* programski jezik koji je uvjerljivo najrašireniji i nudi veliki broj biblioteka za različite primjene. Radno okruženje u kojem je kod pisan je Thonny IDE [17] (eng. *Integrated Development Environment*). Jednostavno je i pregledno sučelje koje nudi sve što je potrebno za efikasno pisanje programskog koda te dolazi s ugrađenim *Python* okruženjem. Omogućava različite funkcije poput vizualizacije memorije, automatskog dovršavanja koda i *Step-by-Step Debugging-a* [6].

Prije početka stvaranja samog programskog koda, potrebno je povezati 40 ulazno/izlaznih priključaka s *Raspebrry Pi* računala s 40 priključaka na PCB pločici upravljačkog sklopa. Za izradu toga su iskorištena dva konektora s 40 pojedinačnih priključaka i 40 žica. Svakoj žici je na krajevima skinut dio izolacije kako bi se mogla zalemiti identično na oba konektora i time povezati jedan priključak s drugim koji se nalazi na istoj poziciji. Na kraju, kako bi se ograničilo pomicanje žica, nanoseno je vruće ljepilo na oba konektora sa spojenim žicama (Slika 5.1.).



Slika 5.1. konektor za povezivanje 40 ulazno/izlaznih priključaka

5.1. Povezivanje GPIO priključaka s upravljačkim sklopom

Kao što je prikazano na električnoj shemi upravljačkog sklopa (Slika 4.1.), nisu svi GPIO priključci iskorišteni. U Tablici 5.1. prikazano je koji su GPIO priključci iskorišteni te za što će biti odgovorni u programskom kodu.

<i>GPIO priključak</i>	<i>Uloga</i>
<i>GPIO 17</i>	Pokretanje desne ruke prema gore
<i>GPIO 27</i>	Pokretanje desne ruke prema dolje
<i>GPIO 22</i>	Pokretanje lijeve ruke prema dolje
<i>GPIO 23</i>	Upravljanje hvataljkom desne ruke
<i>GPIO 24</i>	Pokretanje lijeve ruke prema gore
<i>GPIO 10 (MOSI)</i>	Slanje podataka s <i>Raspberry Pi</i> računala na ADP
<i>GPIO 9 (MISO)</i>	Primanje podataka s ADP-a na <i>Raspberry Pi</i> računalo
<i>GPIO 25</i>	Pokretanje glave u lijevo
<i>GPIO 11 (SCLK)</i>	Sinkronizacija prijenosa podataka između ADP-a i <i>Raspberry Pi</i> računala
<i>GPIO 5</i>	Povezan na priključak za odabir čipa $\overline{CS}/SHDN$
<i>GPIO 16</i>	Pokretanje glave u desno
<i>GPIO 26</i>	Upravljanje hvataljkom lijeve ruke

Tablica 5.1. Povezivanje GPIO priključaka na upravljački sklop

Priključci 1 i 17 daju 3,3V na izlazu te su zajednički povezani vodom na PCB pločici izvedeni kako bi na potrebna mjesta na ostatku sklopa stigao taj napon. Također, priključci 6, 9, 14, 20, 25 i 30 su priključci za masu te su vodom povezani međusobno, a onda i na glavni vod mase na PCB pločici.

5.2. Opis biblioteka korištenih u programskom kodu

Prethodno je, u poglavlju 3.4, opisano kako se inačica 5 *Raspberry Pi* računala razlikuje od prethodnih, posebice inačica 3 i 4. S obzirom da *Raspberry Pi 5* koristi drugačiji čip za GPIO periferne jedinice i zbog toga na drugačiji način mapira iste, jedna od najkorištenijih biblioteka nije kompatibilna s tom inačicom računala. Naime, radi se o biblioteci za upravljanje GPIO perifernim jedinicama *RPi.GPIO* [18] koja je intuitivna i jednostavna za korištenje, a uz to postoji velika količina materijala koju je stvorila zajednica ljudi koji su je stvorili ili su se njome koristili. Iako ova biblioteka nije korištena u konačnom programskom kodu, jer je konačan kod napisan za upotrebu na inačici 5, ipak se u procesu rada koristila i ova biblioteka kada se, zbog određenih problema, testiranje sklopa odvijalo na inačici 3 koja podržava tu biblioteku.

Glavna biblioteka koja se koristila za upravljanje GPIO perifernim jedinicama je *gpio* [19]. Zapravo, to je poveznica za *libgpiod*, Linux biblioteku koja se koristi za upravljanje i interakciju s GPIO periferijama različitih uređaja ne ograničavajući se samo na *Raspberry Pi 5* računalo za razliku od *RPi.GPIO* biblioteke koja je napravljena specifično za rad na *Raspberry Pi* računalima.

Sljedeća važna biblioteka bila je *keyboard* biblioteke [20] koja omogućava interakciju s tipkovnicom, detekciju pritiska i otpuštanja pojedinih tipki, ne blokira glavni program jer se događaji odvijaju u odvojenim dretvama te funkcionira na Windows ili Linux sustavima uz malu razliku. Na Linux sustavima, ova biblioteka zahtijeva naredbu *sudo* koja omogućuje izvršavanje naredbi s *root* privilegijama bez da se korisnik prethodno mora prijaviti kao *root* korisnik. Princip *root* korisnika ključan je u Linux operativnim sustavima te označava glavni korisnički račun koji ima potpuni pristup i ovlasti nad sustavom.

S obzirom da se u upravljačkom sklopu koristi analogno-digitalni pretvornik MCP3008 za očitavanje analognih vrijednosti s potencijometara, u programskom kodu koristile su se dodatne biblioteke specifične namjene. Paket biblioteka *Adafruit-Blinka* [21] sadrži različite biblioteke koje omogućuju korištenje *CircuitPython* biblioteka na Linux uređajima uključujući *Raspberry Pi*. *CircuitPython* je verzija *Python* programskog jezika koji je prilagođen za jednostavno i brzo programiranje, a kao i *Adafruit-Blinka* paket biblioteka razvio ju je Adafruit. Biblioteke

korištene iz paketa su *busio* [21], koji omogućuje interakciju sa sučeljima putem protokola I2C, UART i SPI, *digitalio* [21], koji omogućuje interakciju s digitalnim ulazno/izlaznim priključcima te *board* [21] koji služi za odabir specifičnih priključaka.

Također je korištena posebna biblioteka *adafruit_mcp3xxx.mcp3008* [22] koja je posebno napravljena za rad s analogno-digitalnim pretvornikom MCP3008, iz *adafruit_mcp3xxx* biblioteke uvezena je i funkcija *AnalogIn* koja omogućuje postavljanje kanala ADP-a.

Preostale dvije biblioteke korištene u programskom kodu su standardne biblioteke *time* i *sys*. Biblioteka *time* omogućuje različite funkcije povezane s vremenom [23], dok biblioteka *sys* pruža različite funkcionalnosti za interakciju s *Python* tumačem i operacijskim sustavom [24].

5.3. Programski kod

Funkcionalnost programskog koda objašnjena je na primjeru upravljanja desnom stranom robota, odnosno desnim motorom ruke i desnom hvataljkom. Analogno tome se upravlja i lijevom stranom robota, ali i glavom uz razliku da za dio koda koji se odnosi na glavu robota nema trećeg priključka kao kod ruku s obzirom da ruke osim motora imaju i hvataljku.

5.3.1. Uvoz potrebnih biblioteka

```
# Importing necessary libraries
import sys
import gpiod
import keyboard
import time
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
```

5.3.2. Inicijalizacija GPIO priključaka

Za početak je potrebno inicijalizirati sve GPIO priključke koji će se koristiti u programskom kodu za upravljanje robotom. U Tablici 4 navedeno je koji GPIO priključak odgovoran za koji dio upravljanja robotom te će isto tako biti i u programskom kodu.

```

# chip path
chip_path = 'gpiochip4'

# PIN INITIALIZATION
R_UP = 17
Right_Up = gpiod.Chip(chip_path)
GPIO_Right_Up = Right_Up.get_line(R_UP)
GPIO_Right_Up.request(consumer = "PIN", type=gpiod.LINE_REQ_DIR_OUT)
R_DOWN = 27
Right_Down = gpiod.Chip(chip_path)
GPIO_Right_Down = Right_Down.get_line(R_DOWN)
GPIO_Right_Down.request(consumer = "PIN", type=gpiod.LINE_REQ_DIR_OUT)
R_FIST = 23
Right_Fist = gpiod.Chip(chip_path)
GPIO_Right_Fist = Right_Fist.get_line(R_FIST)
GPIO_Right_Fist.request(consumer = "PIN", type=gpiod.LINE_REQ_DIR_OUT)

```

Naredbom `chip_path = 'gpiochip4'` odabire se čip putem kojega se postavljaju GPIO priključci.

U varijablama `R_UP`, `R_DOWN` i `R_FIST` definirani su brojevi GPIO priključaka koji su odgovorni za pojedinu ulogu. U sljedećem retku se stvara `Chip` objekt, a potom se dohvaća specifična linija, odnosno GPIO priključak te se naredbom `.request(consumer = "PIN", type=gpiod.LINE_REQ_DIR_OUT)` definira da je GPIO izlaznog tipa.

5.3.3. Definiranje početnih stanja GPIO priključaka

```

# DEFINE INITIAL STATES
GPIO_Right_Up.set_value(1)
GPIO_Right_Down.set_value(1)
GPIO_Right_Fist.set_value(1)
GPIO_Left_Up.set_value(1)
GPIO_Left_Down.set_value(1)
GPIO_Left_Fist.set_value(1)
GPIO_Head_Right.set_value(1)
GPIO_Head_Left.set_value(1)

```

S obzirom da se u upravljačkom sklopu koristi invertirajući naponski *buffer*, kod je napisan u negativnoj logici. Zbog toga je na izlazu sklopa visoka razina kada se na GPIO priključke dovede niska razina i obrnuto.

5.3.4. Postavljanje kanala ADP-a

```

# SPI setup
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)
cs = digitalio.DigitalInOut(board.D5) # GPIO5
mcp = MCP.MCP3008(spi, cs)
channel_0 = AnalogIn(mcp, MCP.P0) # Right arm
channel_1 = AnalogIn(mcp, MCP.P1) # Left arm
channel_2 = AnalogIn(mcp, MCP.P2) # Head

```

Koristeći *busio* biblioteku započinje se SPI komunikaciju koja je definirana s 3 priključka (SCK, MISO, MOSI) korištenjem *board* biblioteke. Naredbom `digitalio.DigitalInOut(board.D5)` se odabire i postavlja GPIO priključak (GPIO 5) koji se koristi kao *Chip Select* (CS). U sljedećoj naredbi se stvara *MCP3008* objekt te se odabiru kanali na kojima su spojeni senzori položaja, odnosno potencimetri.

5.3.5. Pretvorba vrijednosti ADP-a

Kao što je navedeno u poglavlju 3.3 (Slika 15) potrebno je pretvoriti digitalnu vrijednost, koju ADP daje na izlazu nakon pretvorbe iz analogne, u kut koji označava položaj ruku i glave. Za taj algoritam pretvorbe potrebno je definirati neke konstante koje će se koristiti kasnije u ostatku programskog koda.

```
# ADC conversion
MAX_angle_arms = 130
MAX_angle_head = 90
# RIGHT ARM
CH0_MAX = 0
CH0_MIN = 0
# LEFT ARM
CH1_MAX = 0
CH1_MIN = 0
# HEAD
CH2_MAX = 0
CH2_MIN = 0
```

Ove varijable su za početak postavljene u nulu, ali se krajnje vrijednosti kasnije postavljaju u odsječku koda za samokalibraciju položaja ruku i glave.

5.3.6. Glavni dio programa

U glavnom dijelu programa vrti se beskonačna *while* petlja te se ispituje je li pritisnuta tipka 'r' na tipkovnici kako bi se aktivirala desna strana robota. Ako jest (tipka mora biti cijelo vrijeme pritisnuta), program nastavlja ispitivati je li pritisnuta tipka 'up', 'down' ili 'f' (eng. *Fist*) te ako je jedna od tipki pritisnuta, odgovarajući GPIO priključak se postavlja u nisko stanje (0V) kako bi na priključak za motor ili hvataljku stigla visoka razina, odnosno napon 24V. Pri izlasku iz dijela gdje je jedna od tipki 'up', 'down' ili 'f' pritisnuta, GPIO priključak se ponovno postavlja u visoku razinu kako ne bi došlo do neželjenih pokreta robota. Program se može prekinuti pritiskom kombinacije tipki 'ctrl' + 'c' nakon čega glavni program prestaje s radom, GPIO priključci se postavljaju u početna stanja kako bi robot mirovao te se priključci otpuštaju naredbom `.release()`.

```

# MAIN PROGRAM
try:
    while True:

        # RIGHT SIDE OF TIOSS ROBOT
        if keyboard.is_pressed('r'):
            if keyboard.is_pressed('up'): # MOVE UP
                while keyboard.is_pressed('up'):
                    GPIO_Right_Up.set_value(0)
                    time.sleep(0.1)
                GPIO_Right_Up.set_value(1)

            if keyboard.is_pressed('down'): # MOVE DOWN
                while keyboard.is_pressed('down'):
                    GPIO_Right_Down.set_value(0)
                    time.sleep(0.1)
                GPIO_Right_Down.set_value(1)

            if keyboard.is_pressed('f'): # CLENCH FIST
                while keyboard.is_pressed('f'):
                    GPIO_Right_Fist.set_value(0)
                    time.sleep(0.1)
                GPIO_Right_Fist.set_value(1)

        else:
            pass

except KeyboardInterrupt:
    print("EXITING...")
finally:
    GPIO_Right_Up.release()
    GPIO_Right_Up.set_value(1)
    GPIO_Right_Down.release()
    GPIO_Right_Down.set_value(1)
    GPIO_Right_Fist.release()
    GPIO_Right_Fist.set_value(1)

```

5.3.7. Algoritam za samokalibraciju položaja ruku i glave

Ideja ovog algoritma je da se robotu daje naredba za samokalibraciju položaja desne ruke, lijeve ruke ili glave kako bi u varijable za krajnje položaje ruku i glave spremio iznos koju daje ADP. Naime, GPIO priključke odgovorne za pokrete ruku prema gore, odnosno za pokrete glave u desno, se postavi u nisku razinu kako bi se ruke kretale prema gore, odnosno glava udesno. Vrijeme kretanja se postavi naredbom `time.sleep(0.1)` koja će dopustiti da se ruke, odnosno glava okreće dovoljno dugo da dođe do mehaničke sklopke koja će isključiti motore. U tom trenutku se očita maksimalna vrijednost krajnjeg položaja i spremi u varijable maksimuma, a onda se isti GPIO priključci vraćaju u visoku razinu, a priključci odgovorni za kretanje ruku prema dolje, odnosno glave prema lijevo se postavljaju u nisku razinu. Na jednaki način se vremenski ograničava trajanje niskog stanja na tim priključcima kako bi ruke i glava došle do krajnjeg položaja i spremile vrijednost

s ADP-a u varijable minimuma te se priključci ponovno vrata u visoko stanje. Time je proces samokalibracije senzora položaja ruku i glave završen i pretvorba vrijednosti ADP-a je omogućena.

```
#POSITION SENSOR SELF-CALIBRATION
sys.stdout.write('SELF-CALIBRATION IN PROCESS')
#DESNA RUKA
GPIO_Right_Up.set_value(0)
time.sleep(15)
CH0_MAX = channel_0.value #MAX vrijednost ADP-a za desnu ruku
GPIO_Right_Up.set_value(1)
GPIO_Right_Down.set_value(0)
time.sleep(15)
CH0_MIN = channel_0.value #MIN vrijednost ADP-a za desnu ruku
GPIO_Right_Down.set_value(1)
#LIJEVA RUKA
GPIO_Left_Up.set_value(0)
time.sleep(15)
CH1_MAX = channel_1.value#MAX vrijednost ADP-a za lijevu ruku
GPIO_Left_Up.set_value(1)
GPIO_Left_Down.set_value(0)
time.sleep(15)
CH1_MIN = channel_1.value#MIN vrijednost ADP-a za lijevu ruku
GPIO_Left_Down.set_value(1)
#LIJEVA RUKA
GPIO_Head_Right.set_value(0)
time.sleep(15)
CH2_MAX = channel_2.value #MAX vrijednost ADP-a za glavu
GPIO_Head_Right.set_value(1)
GPIO_Head_Left.set_value(0)
time.sleep(15)
CH2_MIN = channel_2.value #MIN vrijednost ADP-a za glavu
GPIO_Head_Left.set_value(1)
sys.stdout.write("\r" + " " * 100 + "\r")
sys.stdout.flush()
sys.stdout.write('SELF-CALIBRATION OVER')
```

Samokalibracija se u konačnom kodu pokreće pritiskom kombinaciju tipki 's' + 'c'. Bez pokretanja samokalibracije nije moguće dobiti informacije o položajima ruku i glave s obzirom da su konstante krajnjih položaja postavljene u nulu a prema algoritmu one ne smiju biti jednake nuli (Slika 3.9.). Također je važno napomenuti da je raspon kretanja ruku približno 130°, a raspon okretanja glave je približno 90°. Početni položaj desne ruke je 0°, a krajnji 130°, dok je početni položaj lijeve ruke robota 15°, a krajnji iznosi 155°. Početni položaj glave, odnosno 0° definiran je kada robot gleda ravno naprijed, a to znači da se glava može okretati do -45° u lijevu stranu i do + 45° u desnu stranu.

Konačni kod prilagođen je ovim gore navedenim rasponima i početnim uvjetima. Nakon samokalibracije pri kretanju ruku i glave ispisuje se trenutni položaj pojedinog dijela tijela robota TIOSS.

6. Ispitivanje ispravnosti rada

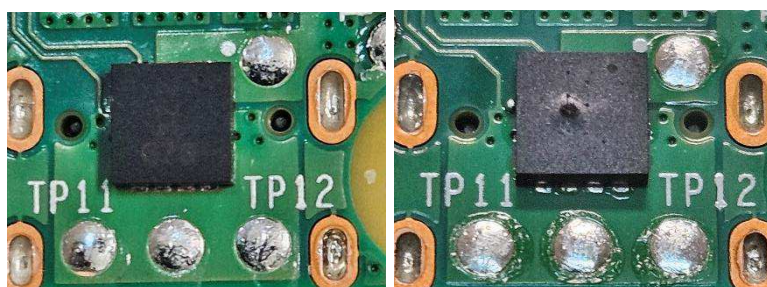
S obzirom da su uz priključke za motore i hvataljke postavljene LED diode koje zasvijetle kada na priključke dođe napon, početno testiranje se moglo odraditi bez povezivanja sklopa s robotom. Prije povezivanja sklopa na izvor napona, multimetrom su ispitane sve točke na sklopu kako bi se provjerilo je li sve ispravno povezano. Nakon toga priključili smo sklop na izvor napajanja 24V, povezali istosmjerni pretvarač napona i *Raspberry Pi 5* računalo. Za početno testiranje je kao napajanje *Raspberry Pi* računala korišteno njegovo napajanje od 27W i s USB-C priključkom. Nakon početnog testiranja sklop je funkcionirao kako je i predviđeno te su vrijednosti s ADP-a bile očitavane.

Sljedeći korak u ispitivanju je bio povezivanje izlaza istosmjernog pretvarača napona, koji na izlazu daje 5V, s priključkom za napajanje *Raspberry Pi* računala.

6.1. Nezgoda pri ispitivanju sklopa

Uslijed priključivanja izlaza pretvarača na priključak za napajanje računala, *Raspberry Pi 5* se nije htio pokrenuti i s vremenom se osjetio pomalo neugodan miris. Bilo je jasno da je nešto pošlo po krivu, ali nikako nije bilo jasno što točno.

Kada se računalo maknulo s ploče na kojem je cijeli upravljački sklop, s donje strane, ispod USB-C priključka za napajanje, pojavila se uočljiva izbočina na čipu (Slika 6.1.). Postalo je jasno da je čip uništen, ali i dalje nije bilo jasno zašto.



Slika 6.1. Čip ispod priključka za napajanje prije i nakon nezgode

Nakon razmišljanja o potencijalnom problemu, na ideju mr. sc Marijana Kurića, postalo je jasno što je uzrokovalo uništenje čipa. Naime u poglavlju 4.1 opisan je istosmjerni pretvarač napona DFR0205 koji na sebi ima sklopku kojom se odabire

hoće li na izlazu pretvarača biti fiksno 5V ili će se potenciometrom namještati izlazni napon u rasponu od 3,3V do 25V. Nažalost, ta sklopka nije čvrsta već je veoma labava te se dogodilo ono najgore, a to je da sklopka nije bila ni u jednom od dva definirana položaja što je uzrokovalo da se na izlazu pretvarača pojavi 24V i na taj način uništi čip *Raspberry Pi 5* računala.

Kako ne bi ponovno došlo do slične nezgode, sklopka je maknuta s pretvarača te se otpornicima fiksiralo da na izlaz pretvarača može doći jedino napon od 5V, a time je onda podignuta razina sigurnosti rada upravljačkog sklopa.

Kako ne bismo oštetili još jedan *Raspberry Pi 5*, u procesu dorade koda i testiranja istog korišten je *Raspberry Pi 3* kojim su se sve iste funkcionalnosti mogle postići iako su se u programskom kodu koristile drugačije biblioteke.

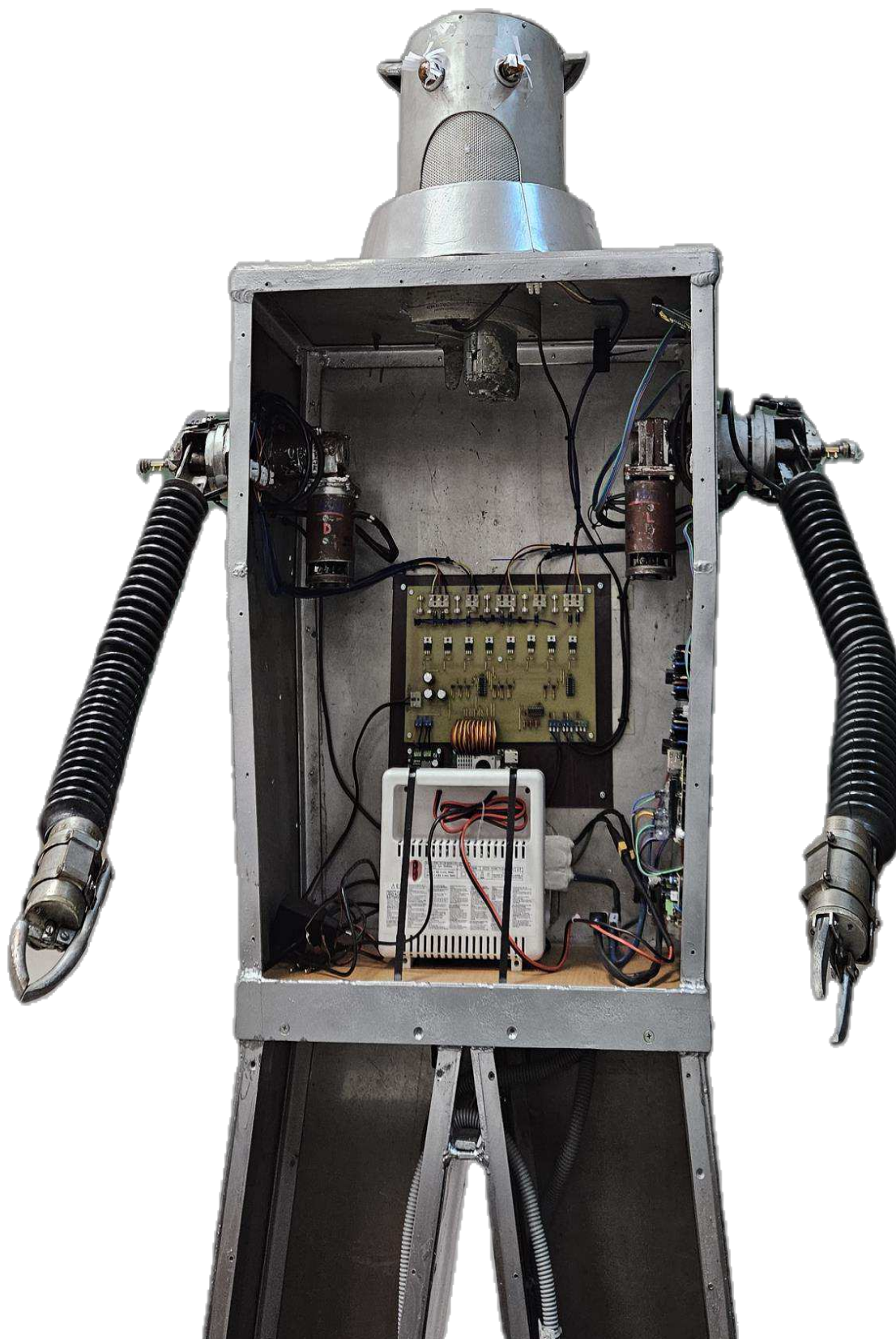
Ipak, s obzirom da je dobar dio ostalih funkcionalnosti robota prilagođena radu na *Raspberry Pi 5* računalu, testiranje sklopa se odradilo i na toj inačici računala. Pri testiranju sklopa, računalo se napajalo samo njegovim napajanjem od 27W (5,1V@5A).

6.2. Testiranje sklopa na robotu TIOSS

Potenciometri su postavljeni na ruke i glavu robota, a ploča s upravljačkim sklopom kojeg čine PCB pločica, *Raspberry Pi 5* i istosmjerni pretvarač napona je pričvršćena za robota s unutrašnje strane leđa robota (Slika 6.2.)

Pri pokušaju povezivanja izlaza pretvarača napona na kojem je 5V i priključka za napajanje *Raspberry Pi* računala došlo je do novoga problema. S napajanjem 5V, a kasnije i isprobanim napajanjem 5,1V kojeg daje pretvarač napona, *Raspberry Pi* se nije uključivao. Pretpostavka je da je problem u kabelu koji se koristi za priključivanje pretvarača na napajanje. Rješenje je, iako ne idealno, korištenje *Raspberry Pi* napajanje od 27W (5,1V@5A) koje je korišteno i pri testiranju sklopa van robota.

Upravljački sklop s programskim kodom je dobro prošao testiranje na robotu TIOSS koje sada ima mogućnost pomicati ruke i hvataljke te rotirati glavom uz konstantnu informaciju o položajima pojedinih dijelova tijela robota u stvarnom vremenu te također može samokalibrirati položaje ruku i glave.



Slika 6.2. Robot TIOSS s ugrađenim upravljačkim sklopom

7. Moguća unaprjeđenja

Fokus ovoga rada je projektiranje i stvaranje upravljačkog sklopa kojim se omogućuje upravljanje rukama robota, hvataljkama i glavom robota. Robotom se upravlja putem *Raspberry Pi* računala, ali ne postoji ograničenje da se samo tim računalom može upravljati robotom. Time je postavljen dobar, kvalitetan i pouzdan temelj za različita unaprjeđenja, ali i implementaciju raznih drugih funkcionalnosti.

Jedna od mogućnosti je korištenje pulsno-širinske modulacije (PWM) kako bi se ruke i glava robota mogle pomicati različitim brzinama. Metoda PWM efikasna je metoda temeljena na izmjeni duljine impulsa signala kojim se upravlja nekim trošilom, u ovom slučaju motorima robota. Promjenom radnog ciklusa (duty cycle) signala kontrolira se koliko dugo traje slanje impulsa što mijenja količinu snage isporučene motorima, a posljedično i brzinu vrtnje motora. Implementacija PWM algoritama, posebice koristeći *Python* na *Raspberry Pi* računalu, nije pretjerano kompleksan zadatak, a dalo bi novu, dodatnu funkcionalnost robotu TIOSS.

U robotu se već nalazi i koristi kamera koja se jednostavno povezuje s *Raspberry Pi* računalom, a pruža različite mogućnosti. Iz tog razloga, upotreba računalnog vida za upravljanje robotom jedna je od naprednijih funkcionalnosti koje se mogu implementirati. Robot, putem kamere, može promatrati svoju okolinu i reagirati na različite načine na različite podražaje. Vrsta podražaja koji bi izazvali reakciju robota mogu biti zaista različiti, ali jedan od mogućih primjena bila bi imitacija ljudskih pokreta. Primjerice, ako osoba ispred robota podigne desnu ruku i robot će podići svoju. Isto tako, robot bi hvataljkom mogao primati objekte koji mu se pružaju kada se, koristeći računalni vid, ti objekti detektiraju u neposrednoj blizini hvataljki robota.

S obzirom da su na ruke i glavu robota postavljeni senzori položaja, odnosno potencijometri, informacije o položaju mogu se koristiti na različite načine. Osnovna primjena bila bi direktno zadavanje položaja pojedinih dijelova tijela robota gdje bi se ruke ili glava robota postavile točno u zadani položaj koristeći kut kao mjeru. Za još preciznije upravljanje motorima može se implementirati i PID (eng. *Proportional-Integral-Derivative*) algoritam koji se bazira na umanjivanju pogreške između željene, ciljane vrijednosti i stvarne vrijednosti koju motori postižu. Uspoređivane vrijednosti bile bi vrijednosti koje daje ADP povezan na potencijometre.

8. Zaključak

Ovim radom restaurirani su i ponovno osposobljeni motori ruku i solenoidi hvataljki robota TIOSS te je dizajniran i napravljen upravljački sklop za upravljanje hvataljkama te motorima ruku i glave. Koristeći *Raspberry Pi* računalo napravljena je i programska podrška za upravljanje robotom putem tipkovnice. Uz to, na ruke i glavu robota postavljeni su senzori položaja kako bi u stvarnom vremenu bila dostupna informacija o trenutnom položaju tih dijelova tijela robota.

Tijekom rada primijenjena su različita znanja poput programiranja u *Python* programskom jeziku, dizajniranje upravljačkog sklopa koristeći znanja iz elektronike, dizajniranje PCB pločice koristeći *Altium Designer* i fizičke izrade pločice. Kroz rad na robotu TIOSS pojavili su se različiti problemi i poteškoće, ali su se stalozanim i temeljitim pristupom pronašla rješenja za te probleme.

Robot TIOSS ponovno može pomicati ruke, koristiti hvataljke i okretati glavu putem upravljačkog sklopa opisanog u ovom radu. Time je postavljen sjajan temelj za buduća unaprjeđenja i implementaciju dodatnih funkcionalnosti robota poput korištenja pulsno-širinske modulacije za upravljanje motorima ili implementacije računalnog vida i povezivanje istog s upravljanjem kretanja robota.

Konačno, ovaj rad je samo još jedan korak u obnovi i modernizaciji robota TIOSS, prvog humanoidnog robota u Hrvatskoj, simbola samih početaka robotike u Hrvatskoj.

9. Literatura

- [1] Životni intervju: Branimir Makanec, <https://www.vecernji.hr/vijesti/moj-robot-robi-je-68-bio-prava-senzacija-setao-se-zagrebackim-ulicama-i-dobacivao-curama-1447912>, pristupljeno 15. lipnja 2024
- [2] Robi spašen s otpada, Prvi hrvatski robot sreo 'tatu' nakon 60 godina: 'Kakav dar!', <https://www.24sata.hr/lifestyle/moj-robot-je-po-zagrebu-hodao-jos-1958-robi-je-bio-pravi-hit-693180>, pristupljeno 15. lipnja 2024.
- [3] *Modernizacija kretanja robota TIOSS*, Marko Tomić, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, lipanj 2023.
- [4] *Modernizacija kretanja robota TIOSS korištenjem robotskog vida i govora*, Jana Alajbeg, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, lipanj 2023.
- [5] *MUR420 datasheet*, *alldatasheet.com*, <https://www.alldatasheet.com/datasheet-pdf/pdf/678492/KEXIN/MUR420.html>, pristupljeno 28. travnja 2024.
- [6] *IRLZ44N datasheet*, *alldatasheet.com*, <https://www.alldatasheet.com/datasheet-pdf/pdf/68872/IRF/IRLZ44N.html>, pristupljeno 28. travnja 2024.
- [7] Službena dokumentacija *Raspberry Pi*, <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>, pristupljeno 25. travnja 2024.
- [8] *BS 170 datasheet*, *alldatasheet.com*, <https://www.alldatasheet.com/datasheet-pdf/pdf/2967/MOTOROLA/BS170.html>, pristupljeno 28. travnja 2024
- [9] *CD74HC4049 datasheet*, *alldatasheet.com*, <https://www.alldatasheet.com/datasheet-pdf/pdf/27055/TI/CD74HC4049.html>, pristupljeno 2. svibnja 2024.
- [10] Position Sensor – an overview, <https://www.sciencedirect.com/topics/engineering/position-sensor>, pristupljeno 15. svibnja 2024.
- [11] *MCP3008 datasheet*, *alldatasheet.com*, <https://www.alldatasheet.com/datasheet-pdf/pdf/92963/MICROCHIP/MCP3008.html>, pristupljeno 15. svibnja 2024.
- [12] Raspberry Pi računalo kroz povijest,

- <https://opensource.com/resources/raspberry-pi> , pristupljeno 4. lipnja 2024.
- [13] *Altium Designer*, <https://www.altium.com/altium-designer> , pristupljeno 20. svibnja 2024.
- [14] *Altium Designer općenito*, https://en.wikipedia.org/wiki/Altium_Designer , pristupljeno 19. lipnja 2024.
- [15] *DFR02005 istosmjerni pretvarač napona*, <https://www.dfrobot.com/product-752.html>, pristupljeno 20. lipnja 2024.
- [16] *Upute za izradu tiskane pločice, Organizacija rada u studentskim laboratorijima*, https://www.zesoi.fer.hr/zesoi/obrazovanje/nastavni_laboratoriji , pristupljeno 10. lipnja 2024.
- [17] *Thonny IDE*, <https://thonny.org/> , pristupljeno 20. lipnja 2024.
- [18] *RPi.GPIO Python biblioteka* <https://pypi.org/project/RPi.GPIO/#description>, pristupljeno 21. lipnja 2024.
- [19] *gpiod poveznica za libgpiod Python biblioteku* <https://pypi.org/project/gpiod/>, pristupljeno 15. svibnja 2024.
- [20] *keyboard Python biblioteka*, <https://pypi.org/project/keyboard/> , pristupljeno 15. svibnja 2024.
- [21] *Adafruit-Blinka Python biblioteka*, <https://pypi.org/project/Adafruit-Blinka/> , pristupljeno 21. svibnja 2024.
- [22] *adafruit_mcp3xxx.mcp3008 Python biblioteka*, <https://docs.circuitPython.org/projects/mcp3xxx/en/latest/> , pristupljeno 21. svibnja 2024.
- [23] *time Python biblioteka*, <https://docs.Python.org/3/library/time.html>, pristupljeno 15. svibnja 2024.
- [24] *sys Python biblioteka*, <https://docs.Python.org/3/library/sys.html>, pristupljeno 15. svibnja 2024.

Sažetak

UPRAVLJANJE RUKAMA HUMANOIDNOG ROBOTA TIOSS

Ovaj rad dio je obnove i modernizacije robota TIOSS, prvog humanoidnog robota u Hrvatskoj koji je nastao još šezdesetih godina 20. stoljeća. Glavni cilj rada bio je osposobljavanje motora ruku i hvataljki koje funkcioniraju na principu solenoida. Dizajniran je i napravljen upravljački sklop sastavljen od PCB pločice, istosmjernog pretvarača napona i *Raspberry Pi* računala. Centar upravljačkog sklopa je dizajnirana PCB pločica kojom se koristeći N-kanalne MOSFET-e kao sklopke povezuje motore ruku, hvataljke, ali i glavu robota s *Raspberry Pi* računalom. Koristeći programski jezik *Python* napravljena je i programska podrška za upravljački sklop koja je prilagođena za inačicu 5 *Raspberry Pi* računala. Osim upravljanja hvataljkama te motorima ruku i glave, koristeći potenciometre kao senzore položaja omogućeno je praćenje položaja navedenih dijelova tijela robota u stvarnom vremenu.

Ključne riječi: robot TIOSS, upravljački sklop, solenoid, motori, ruke, hvataljke, glava, MOSFET, *Raspberry Pi*, senzor položaja, potenciometar, PCB, *Python*, samokalibracija

Summary

CONTROL OF THE HANDS OF THE HUMANOID ROBOT TIOSS

This work is part of the renovation and modernization of the TIOSS robot, the first humanoid robot in Croatia created in the 1960s. The main goal of the work was to enable the functioning of the arm motors and grippers, which operate on the principle of solenoids. A control circuit consisting of a PCB board, a DC voltage converter, and a *Raspberry Pi* computer was designed and built. The central part of the control circuit is the designed PCB board, which uses N-channel MOSFETs as switches to connect the arm motors, grippers, and the robot's head to the *Raspberry Pi* computer. Using the *Python* programming language, software for the control circuit was also created and adapted for version 5 of the *Raspberry Pi* computer. In addition to controlling the grippers and the motors of the arms and head, potentiometers are used as position sensors to enable real-time tracking of the positions of the mentioned body parts of the robot.

Keywords: TIOSS robot, control circuit, solenoid, motors, arms, grippers, head, MOSFET, *Raspberry Pi*, position sensor, potentiometer, PCB, *Python*, self-calibration