

# Robotska ruka upravljana rukavicom

---

**Kesić, Kristijan**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:232985>

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

*Download date / Datum preuzimanja:* **2024-12-30**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Stručni studij**

**ROBOTSKA RUKA UPRAVLJANA RUKAVICOM**

**Završni rad**

**Kristijan Kesić**

**Osijek, 2023.**

# SADRŽAJ

<i>1. Uvod</i> .....	<i>1</i>
<i>2. Teorijski osvrt na robotsku ruku</i> .....	<i>2</i>
2.1 Teorija problema.....	3
2.2 Prijedlog sklopovskog rješenja .....	3
2.3 Prijedlog algoritamskog rješenja.....	4
2.4 Prijedlog upravljačkog i komunikacijskog sučelja .....	4
<i>3. Realizacija sustava</i> .....	<i>5</i>
3.1 Korišteni alati i programska sučelja .....	5
3.2 Izrada upravljačke rukavice .....	5
3.3 Izrada robotske ruke .....	8
3.4 Realizacija algoritamskog rješenja.....	12
3.5 Realizacija sklopovskog rješenja .....	15
<i>4. Testiranje sustava</i> .....	<i>19</i>
4.1 Metode i postupci testiranja.....	19
4.2 Rezultati testiranja.....	19
<i>5. Zaključak</i> .....	<i>24</i>
<i>LITERATURA</i> .....	<i>25</i>
<i>SAŽETAK</i> .....	<i>26</i>
<i>ABSTRACT</i> .....	<i>27</i>
<i>ŽIVOTOPIS</i> .....	<i>28</i>
<i>PRILOZI</i> .....	<i>29</i>

## 1. .Uvod

Robotika objedinjuje niz znanstvenih područja i disciplina, kao što su elektrotehnika, elektronika, strojarstvo, automatika, računalstvo i područje umjetne inteligencije. U primjeni su danas najviše industrijski roboti, za kojima je velika potražnja u automobilskoj industriji i kao sredstva unutarnjeg transporta. Brzina, velika preciznost, točnost, mogućnost ponavljanja pokreta su glavne prednosti robota.

Robotske ruke mogu zamijeniti ljudske u okolinama koje su opasne za ljudsko zdravlje (npr. radijacija, vakuum, plinovi koji su opasni za dišni sustav ljudi). Također se mogu koristiti u teško pristupačnim lokacijama kao što su svemir, morske dubine, ruševine zgrada.

Ovim završnim radom opisano je projektiranje, izrada i programiranje robotske ruke upravljane upravljačkom rukavicom. Ovaj rad prikazuje jednostavnu verziju humanoidnih robotskih ruku upravljani imitacijom ljudske ruke. Naprednija verzija rada bi se mogla koristiti u područjima koja su opasna za ljude (radioaktivna područja, vakuum, atmosfera nepogodna za boravak čovjeka) ili nedostupna i teško dostupna (ispod vode, udaljena područja, svemir).

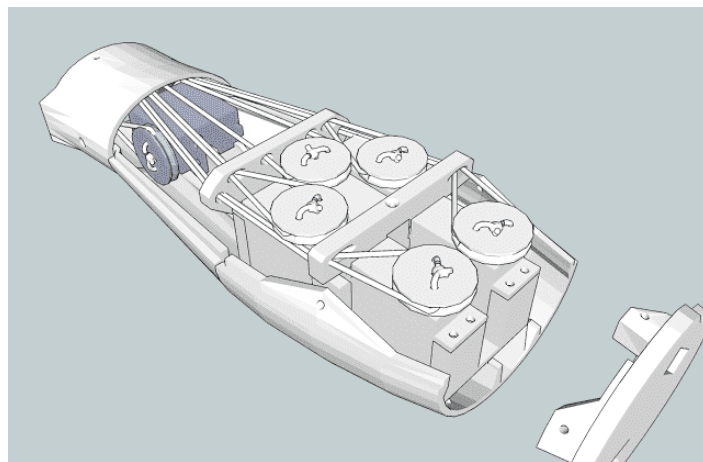
Cilj je izraditi i programirati model robotske ruke upravljane upravljačkom rukavicom. Ruka treba imitirati položaj rukavice. Svaki prst treba imati mogućnost gibanja neovisno o poziciji drugih prstiju. Prsti se mogu saviti u člancima, a dlan se može saviti u zglobu. Rukavica se treba povezati s rukom bežično pomoću Wi-Fi signala.

Preko objašnjenja teorijskog osvrta i prijedloga algoritamskog rješenja te upravljačkog i komunikacijskog sučelja objašnjeno je kojim alatima i programskom sučeljima se koristilo u izradi robotske ruke.

U posljednjem poglavlju поближе je opisano testiranje sustava, kao i metode, postupci korišteni u testiranju i rezultati dobiveni prilikom testiranja rada robotske ruke.

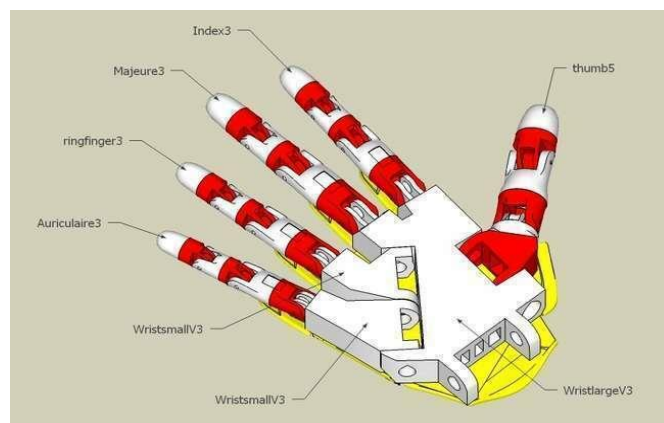
## 2. Teorijski osvrt na robotsku ruku

Struktura ruke koja je korištena je InMoov [1] robotska ruka koja je napravljena od PLA materijala. Ruka je isprintana na 3D printeru. Cilj je da se prsti pomiču pomoću sustava remenica. Na jednom koturu se nalazi namotana žica koja je provedena kroz prst, te pri okretanju kotura u jednu ili drugu stranu prst se savija ili izravnavava. Svaki prst ima svoj motor na koji je spojen kotur. To omogućuje svakom prstu da se individualno pomiče u odnosu na poziciju drugih prstiju. Motori i koturi se nalaze u podlaktici ruke. Na slikama 1.1 i 1.2 je prikazan 3D model [2] robotske ruke.



Sl. 1.1 Prikaz podlaktice ruke i motora u podlaktici ruke

Ruka je kreirana u više manjih dijelova, na taj način ruka može imati složenije pokrete.



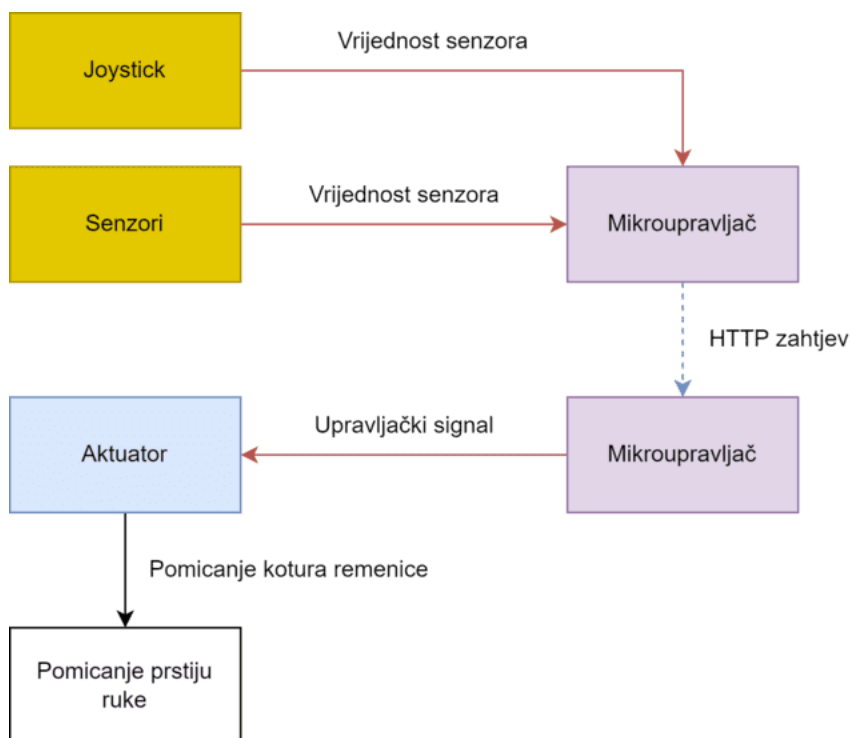
Sl. 1.2 Prikaz dijelova ruku

## 2.1 Teorija problema

Ovaj rad prikazuje jednostavnu verziju humanoidnih robotskih ruku upravljani imitacijom ljudske ruke. Prsti se mogu saviti u člancima, a šaka se može rotirati u zglobu. Naprednija verzija rada bi se mogla koristiti u područjima koja su opasna za ljude (radioaktivna područja, vakuum, atmosfera nepogodna za boravak čovjeka) ili nedostupna i teško dostupna (ispod vode, udaljena područja, svemir).

## 2.2 Prijedlog sklopovskog rješenja

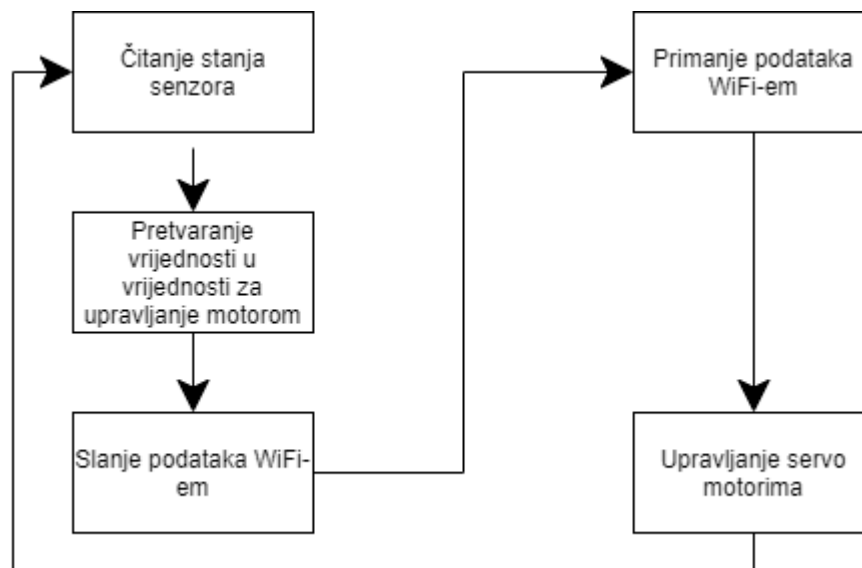
Rukavica ima senzore savijanja (*eng. flex sensor*) [4] koji detektira kada je prst savijen i koliko je savijen. Ukupno je na rukavici pet takvih senzora, jedan za svaki prst. Na taj način se svaki prst može individualno upravljati. Mikroupravljač ESP8266 [5] prima signale sa senzora i šalje ih pomoću *WiFi-a* koji komunicira s istim modulom koji je spojen na ruci. Motorima u robotskoj ruci upravlja mikroupravljač ESP8266 prima signale s rukavice i upravlja motorima MG996R [3] u ruci. Za upravljanje rotacijom zgloba koristi se analogni *joystick*, ali samo X-os. Na slici 1.3 je prikazan prijedlog sklopovskog rješenja.



Sl. 1.3 Shema prijedloga sklopovskog rješenja

## 2.3 Prijedlog algoritamskog rješenja

Senzori trebaju detektirati poziciju prstiju u rukavici, te nakon toga vrijednosti koje očitaju sa senzora i *joystick*-a obradi mikroupravljač na rukavici. Te veličine mikroupravljač pošalje preko *WiFi* modula mikroupravljaču na ruci koji pomoću tih veličina upravlja motorima. Na slici 2.1 je prikazan prijedlog algoritamskog rješenja.



Sl. 2.1 Shema prijedloga algoritamskog rješenja

## 2.4 Prijedlog upravljačkog i komunikacijskog sučelja

Upravljanje rukom vrši se upravljačkom rukavicom koja je spojena *WiFi* modulom s robotskom rukom. Korišteni *WiFi* modul je ESP8266. Klijent modul koji se nalazi na rukavici i šalje podatke, a server modul koji se nalazi na modelu robotske ruke i prima podatke. Server stvara SoftAP (*eng. software enabled access point*) tako da se *WiFi*jem može direktno na njega spojiti, te pokreće web server koji na zahtjev uz potrebne podatke očitanih senzora ispisuje te vrijednosti na servo motore.

## 3. Realizacija sustava

### 3.1 Korišteni alati i programska sučelja

Za pokretanje ruke koriste se motori MG996R, za senzore koji su na rukavici koriste se savitljivi (*eng. flex*) senzori ZD10-100. Struktura ruke je 3D printana na Ender 3 3D printeru, a za materijal ruke se koristiti materijal PLA. Mikroupravljač koji se koristiti je ESP8266 koji se nalazi u ruci i upravlja pokretima motora. Programiran je u PlatformIO-u, koristeći Arduino biblioteku. Za bežičnu komunikaciju između mikroupravljača koristi se ESP8266.

### 3.2 Izrada upravljačke rukavice

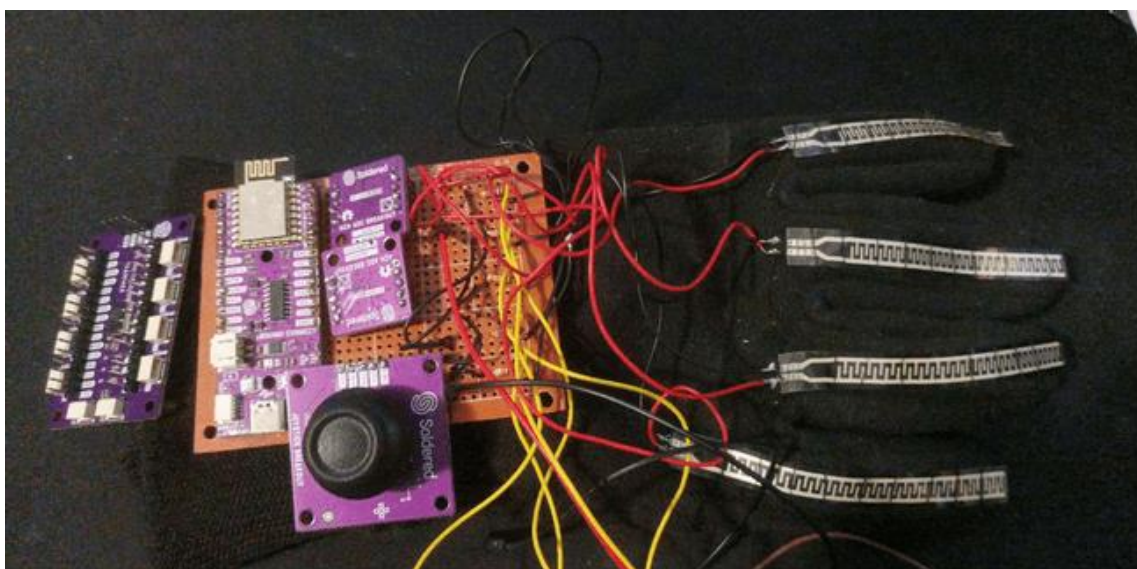
Za izradu upravljačke rukavice potrebno je izabrati rukavicu koja je dovoljno savitljiva da se senzori mogu savijati oko nje, ali i dovoljno čvrstu da se ostale komponente na njoj ne miču. Zbog tog razloga korištena je zimska biciklistička rukavica. Dodatna je prednost odabrane rukavice to što se senzori mogu zašiti za rukavicu. Na senzore je prije šivanja potrebno zalemiti vodiče koji se koriste za povezivanje s ostatkom komponenti upravljačke rukavice. Senzore su zašiveni za rukavicu kevlarskim koncem. Potrebno ih je zašiti na dnu prsta da ne klize prema šaci. Na slici 3.1 je prikazana rukavica na koju su zašiveni senzori, a na senzore su zalemljeni vodiči. Slika 3.2 prikazuje komponente upravljačke rukavice povezane sa sensorima na upravljačkoj rukavici.





Sl. 3.1 Upravljačka rukavica sa senzorima

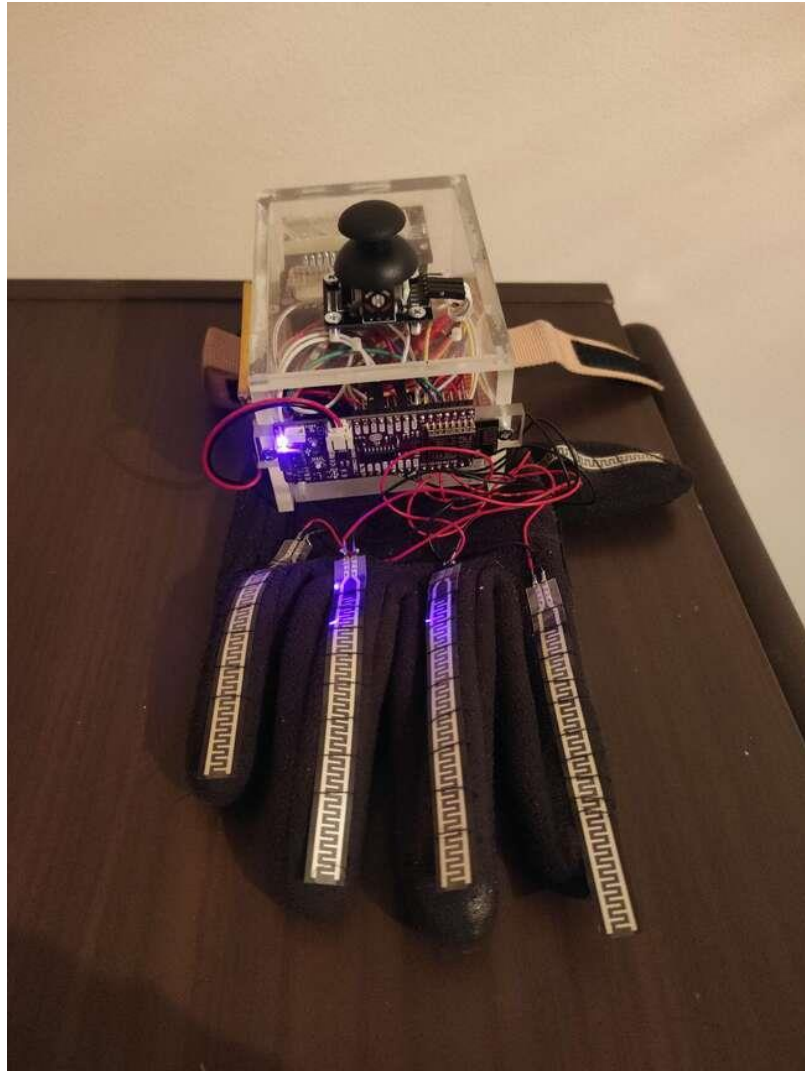
Nakon toga postavi se upravljačke komponente na bušenu pločicu i zalemiti ih. Vodiči senzora se zaleme prema shemi spoja rukavice.



Sl. 3.2 Senzori zalemljeni na bušenu pločicu prema shemi

Da bušena pločica bude sigurno učvršćena za rukavicu koristi se postolje napravljeno od polikarbonata koje je spojeno remenom oko rukavice. Probušene su rupe kroz postolje u

ravnini s rupama u bušenoj pločici te su spojene vijkom, a matica se koristi radi držanja udaljenosti postolja i bušene pločice. U postolju su također napravljene rupe kroz koje se provuče remen kojim se veže upravljačka kutija oko ruke, a remen se lijepi čičkom. Potrebno je zaštititi komponente od vanjskih utjecaja. Zbog tog razloga kreirano je kućište za komponente na rukavici. Na slici 3.3 prikazana je rukavica s kućištem za komponente.



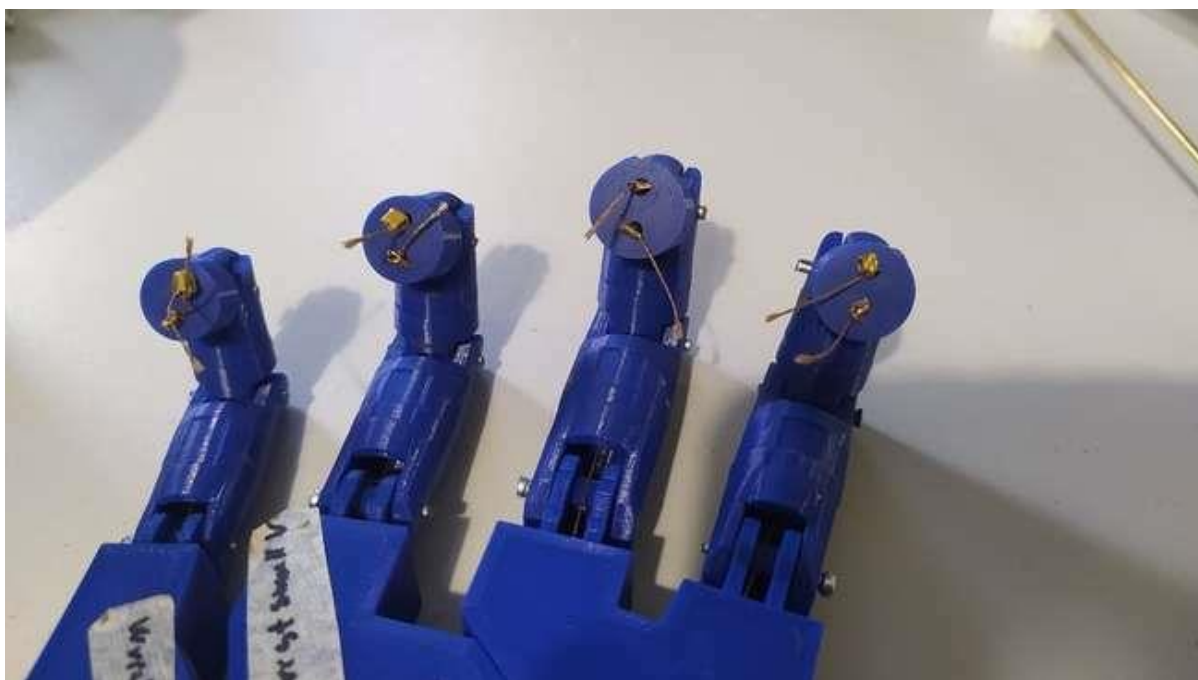
Sl. 3.3 Rukavica s kućištem za komponente





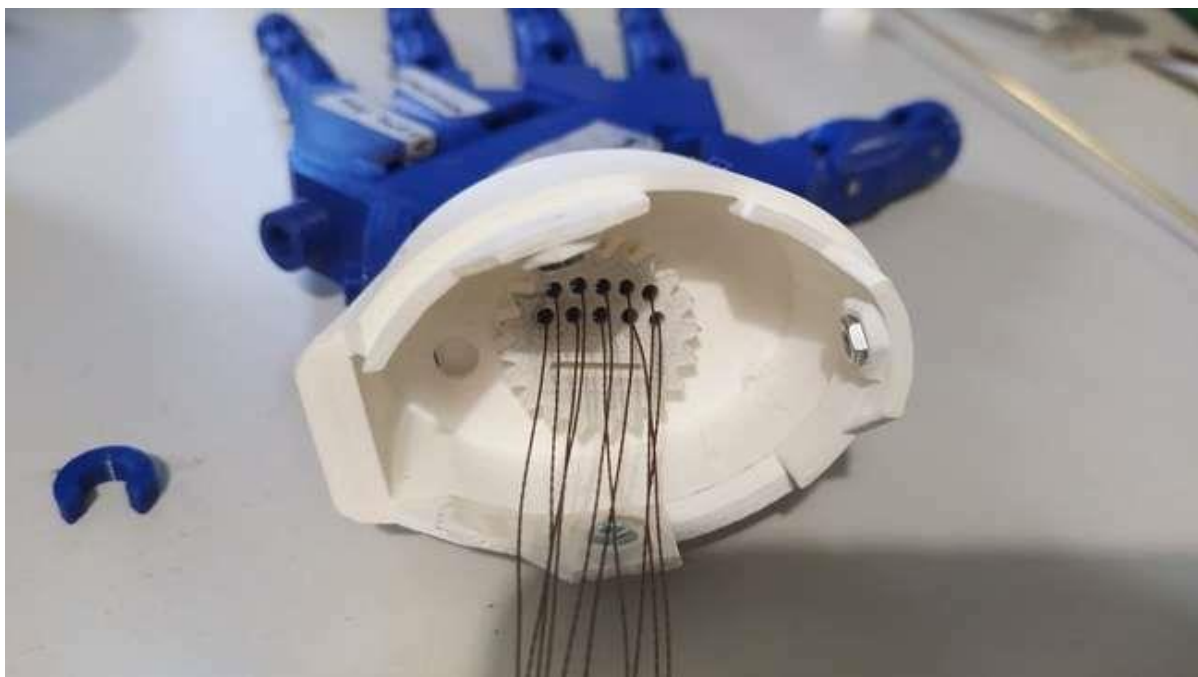
Sl. 3.5 Motori u podlaktici ruke

Kevlarski konac se provuče kroz rupe u prstima te kroz rupe u bazi šake. Vrhovi konca se veću za mali komad žice od mesinga. Na slici 3.6 prikazan je način na koji je konac spojen na vrhu prstiju.



Sl. 3.6 Vrhovi prstiju rubom od mesinga

Sastavljen je zglob ruke te je kroz njega provučen konac. Na slici 3.7 je prikazan način kako je provučen konac kroz zglob robotske ruke.



Sl. 3.7 Konac na dnu robotske ruke kako prolazi kroz zglob

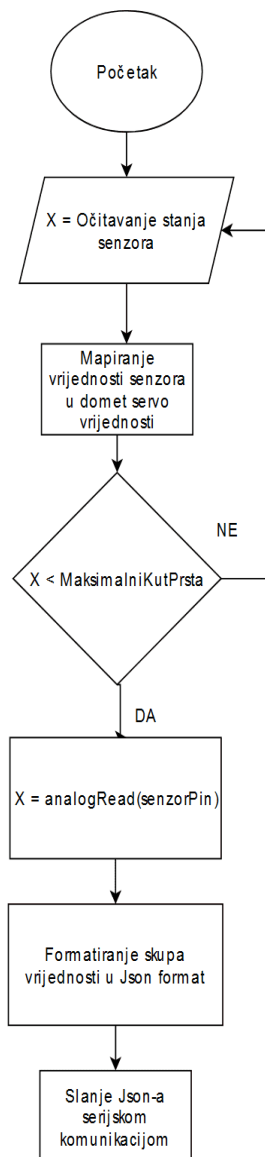
Kako bi konac bio nategnut postavljene su opruge koje osiguravaju da su konci u prstu uvijek nategnuti, a kroz opruge su stavljene gumene cijevi kako bi se smanjilo trenje između konca i opruge. Na kraju je konac spojen na remenice. Na slici 3.8 prikazana je složena robotska ruka.



sl. 3.8. Sastavljena robotska ruka

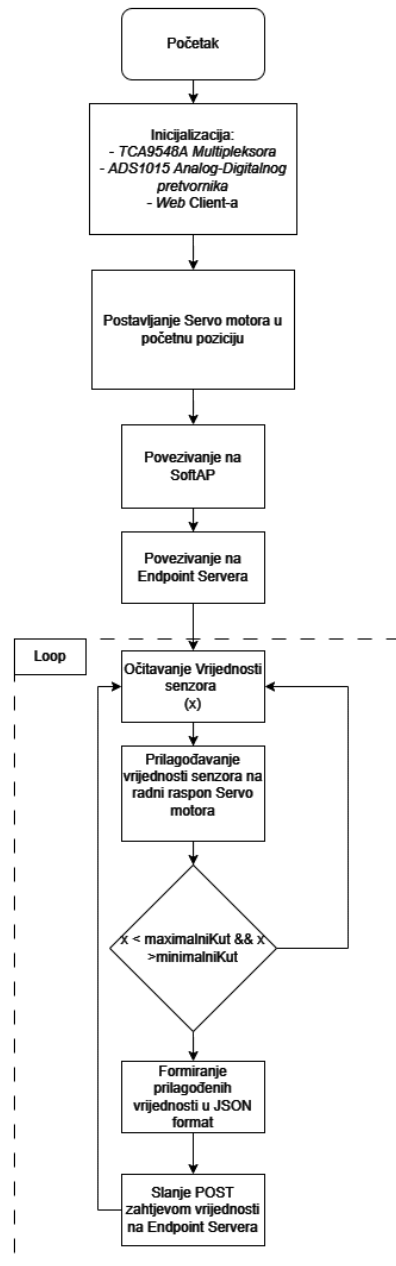
### 3.4 Realizacija algoritamskog rješenja

Upravljanje robotskom rukom činimo upravljačkom rukavicom, a vezu između upravljačke rukavice i robotske ruke postizemo WiFi-em. Svaki mikroupravljač potrebno je posebno programirati i potrebno je uspostaviti komunikaciju između rukavice i ruke. Tok rada upravljačke rukavice je prikazan na slikama 3.9. i 3.10.



Sl. 3.9 Očitavanje vrijednosti senzora

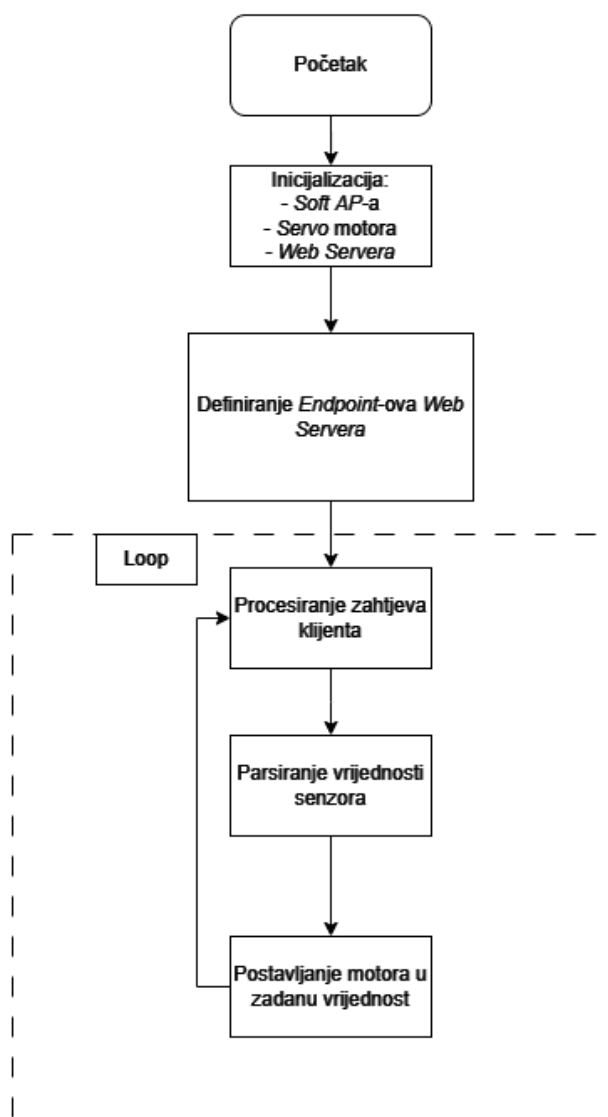
Na slici 3.9 prikazan je dijagram koji prikazuje kako se čitaju vrijednosti iz senzora i slanje vrijednosti serijskom komunikacijom na modul koji se koristi za Wi-Fi komunikaciju.



Sl. 3.10 Dijagram toka upravljačke rukavice

Na slici 3.10 prikazan je dijagram koji opisuje način rada upravljačke rukavice.

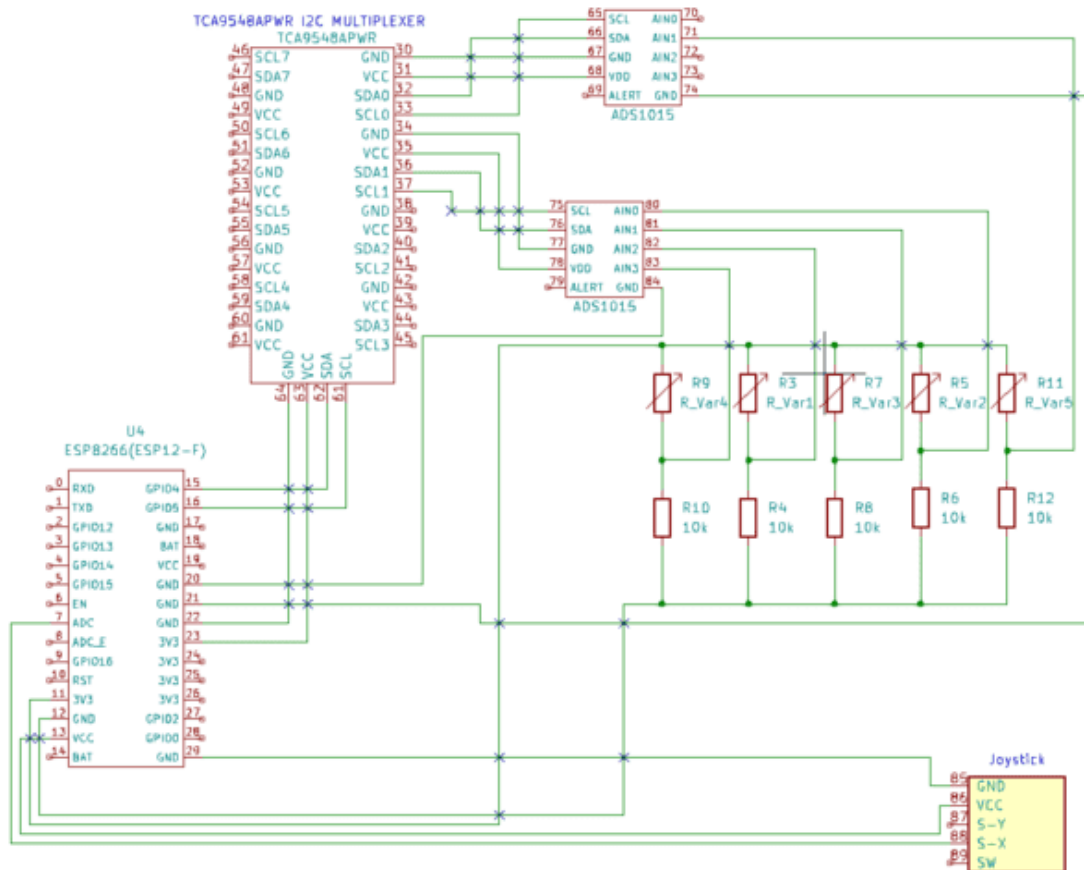




Sl. 3.11 Dijagram toka robotske ruke

Na slici 3.11 prikazan je dijagram rada motora u robotskoj ruci. Primaju vrijednosti s rukavice, te postavljaju motore u zadanu poziciju.

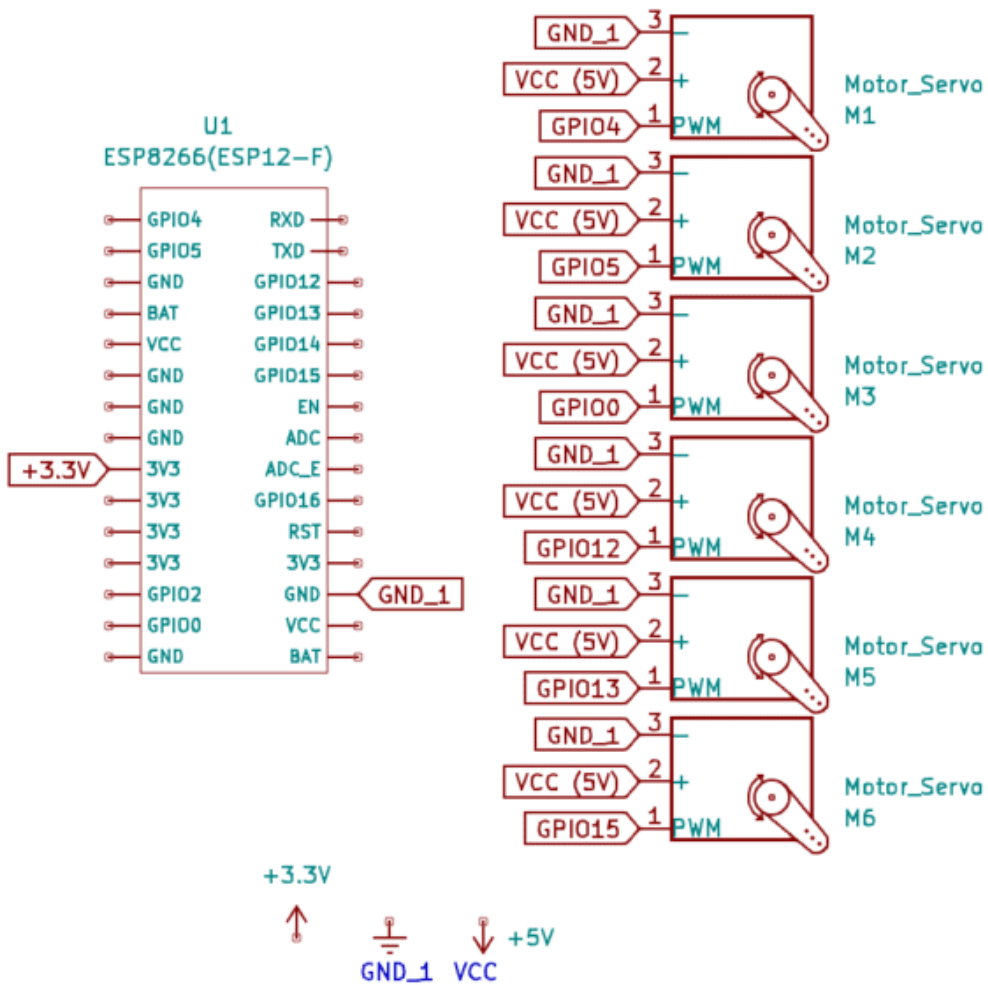
### 3.5 Realizacija sklopovskog rješenja



Sl. 3.12 Shema spoja rukavice

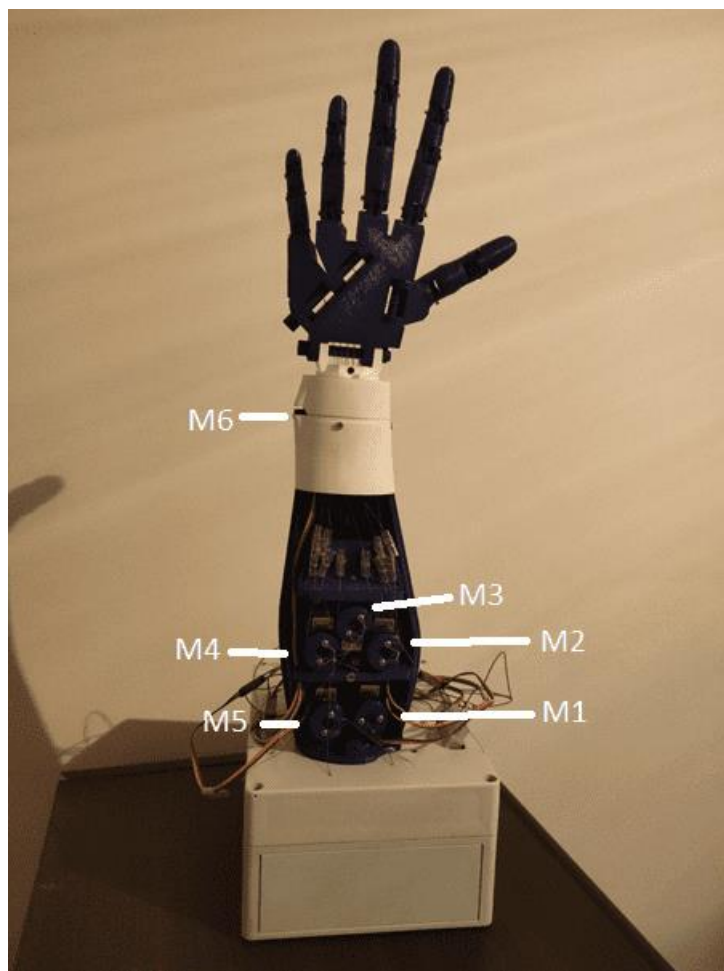
Na slici 3.12 je prikazana shema rukavice. Svaki od senzora je spojen na otpornik od  $10\text{k}\Omega$  u seriju kako bi se povećao raspon vrijednosti koje može vratiti. Senzori su spojeni na analogne pinove. Na mikroupravljač je spojen multiplexer TCA9548A [6] koji omogućava komunikaciju više uređaja s mikroupravljačem. To postiže kreiranjem pojedinih kanala, pri čemu čita vrijednosti s odabranog kanala i blokira ostale kanale. Na multiplexer TCA9548A spojena su dva dodatna ADS1015 [7] modula koji su također multiplexori. Svaki ADS1015 ima 4 kanala, sa svakog kanala očita napon i pretvara ju u digitalnu vrijednost, u ovome slučaju koristi se multiplexer jer nije bilo dovoljno SDA (*Serial Data*) i SDC (*Serial Dana Clock*) pin-ova. Preko I2C (*inter-integrated circuit*) komunikacije mikroupravljač podešava analogno-digitalni pretvarač, te očitane vrijednosti pretvara u prethodno podešene

digitalne signale. Za upravljanje rotacijom šake koristi se *joystick*, koji je također spojen na ADS1015.

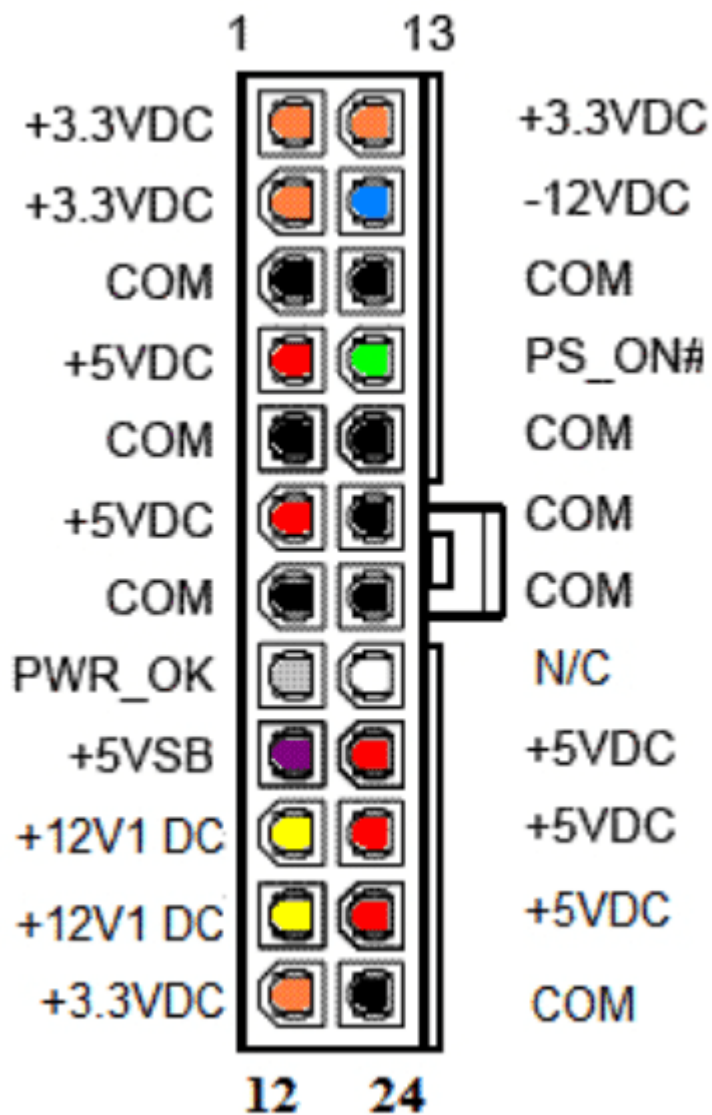


Sl. 3.13 Shema spoja robotske ruke i napajanja

Na slici 3.13. je prikazana shema spajanja motora na upravljački modul i napajanja motora. Slika 3.14 prikazuje položaj motora i njihovu oznaku iz sheme u modelu robotske ruke. Motori su napajani napajanjem koje se koristi u računalu. Kako bi se računalno napajanje koristilo bez matične ploče potrebno je spojiti zelenu žicu označenu PS\_ON (pin 16) na slici 3.15 s bilo kojom drugom žicom uzemljenja označenom crnom bojom.



Sl. 3.14 Snimak robotske ruke s označenim prikazom položaja motora



Sl. 3.15 Konektor s 24 pina na računalnom napajanju

Za napajanje motora potrebno je koristiti napajanje od 4.8V do 7.2V. Zbog toga se koristi napajanje od 5V označeno crvenom žicom, te se koristi uzemljenje označeno crnom žicom.

## 4. Testiranje sustava

### 4.1 Metode i postupci testiranja

U ovom poglavlju opisan je način testiranja upravljačke rukavice, robotske ruke i komuniciranje pomoću *WiFi*-a. Potrebno je testirati i ograničiti kut rotacije motora da prije nego što oštetimo model ruke. Također je potrebno testirati komunikaciju između upravljačke rukavice i robotske ruke.

U prvom dijelu se testira upravljanje motorima i granice do kojih motori mogu doći bez da oštete model ruke, a u drugom dijelu se testira preciznost pomicanja ruke.

### 4.2 Rezultati testiranja

Testiran je raspon *servo* motora korištenjem serijske komunikacije kroz upravljački terminal (*Serial Monitor-a*), kako bi se odredio granični kut kojeg kevlarški konac može izdržati, radi prevencije oštećenja opreme. Također se mora postaviti maksimalni kut rotacije *servo* motora. Određivanje maksimalnog kuta *servo* motora se postigne postavljanjem prstiju u krajnju ispravljenu, a zatim u krajnju savijenu poziciju modela prsta, te pomoću upravljačkog terminala i vijaka kotura na *servo* motoru. Potrebno je promatrati napetost niti. Taj proces se ponavlja za svaku krajnju poziciju prsta modela ruke, to jest *servo* motora. Na slici 4.1 je prikazan donji dio podlaktice robotske ruke u kojem su učvršćeni *servo* motori.



Sl. 4.1 Slika postavljenih remenica na motorima

Senzore savitljivosti na upravljačkoj rukavici se testiraju tako da ih se spoji na analogno-digitalni pretvarač ADS1015. Na mikroupravljač ESP8266 je spojen multipleksor TCA9548A koji vrši I2C komunikaciju s mikroupravljačem te s trenutno otvorenog kanala očitava vrijednosti spojenog analogno-digitalnog pretvarača ADS1015 u mikroupravljač, te zatim zatvara trenutni kanal, prebacuje se na idući zadani kanal te ponovo vrši očitavanje s otvorenog kanala, radi nedostatka analognih pin-ova na mikroupravljaču ESP8266 potrebno je koristiti analogno-digitalni pretvarač, u ovome slučaju ADS1015. Mikroupravljač ESP8266 se spoji s računalom i preko upravljačkog terminala očitava vrijednosti senzora. Slika 4.2 prikazuje vrijednosti snimljene dok je senzor (prst rukavice) ispružen. Slika 4.3 prikazuje vrijednosti snimljene kada je senzor (prst rukavice) savijen.

Palac	Kažiprst	Srednji prst	Prstenjak	Mali prst	
111	0	0	0	0	Ispružen prst
156	0	0	0	0	
211	0	0	0	0	
0	0	0	0	0	
0	0	0	0	0	
0	0	1	0	0	
108	0	0	0	0	
0	88	887	114	0	
0	0	0	0	0	
0	0	0	0	0	
58.6	8.8	88.8	11.4	0	Srednja vrijednost
70.32	15.84	159.64	20.52	0	Prosječno odstupanje

Sl. 4.2 Rezultati mjerenja senzora ispruženog prsta

Palac	Kažiprst	Srednji prst	Prstenjak	Mali prst	
698	679	1047	729	1034	Savijen prst
709	648	1043	694	1033	
680	693	1046	717	1040	
695	690	1043	715	1036	
699	690	1043	717	1036	
699	687	1043	714	1038	
695	683	1043	703	1038	
701	682	1042	702	1035	
697	681	1041	699	1040	
683	670	1041	672	1034	
696	680.3	1043.2	706.2	1036.4	Srednja vrijednost
5.88	8.78	1.32	12.2	2.08	Prosječno odstupanje

Sl. 4.3 Rezultati mjerenja senzora savijenog prsta

Potrebno je ograničiti *servo* motor kako u ispruženom stanju ne bi pukao kevlarski konac koji povlači prste modela u ispruženu poziciju. To se postigne postavljanjem prsta u ispruženu poziciju, te zatezanjem konca na vijku kotura *servo* motora. Zatim se *servo* motor postavlja u kutove bližima graničnoj savijenoj vrijednosti modela ruke, dok konac ne bude dovoljno zategnut da se može micati, ali ne dovoljno zategnut da pukne. Na slici 4.4 ograničenje motora je iskazano u stupnjevima .



Palac	Kažiprst	Srednji prst	Prstenjak	Mali prst	
0	0	0	0	0	Donja granica
100	155	155	155	100	Gornja granica

Sl. 4.4 Ograničenje motora

Testiranje komunikacije se provjerava na ESP8266 modulu na rukavici. Program mikroupravljača rukavice ispisuje vrijednosti senzora savitljivosti na rukavici, te status slanja podataka. Podatke poslane s upravljačke rukavice prima mikroupravljač na robotskoj ruci. Motori u modelu robotske ruke se rotiraju u zadani kut. Mikroupravljač rukavice se povezuje na predefiniranu žarišnu točku koju mikroupravljač ESP8266 na modelu robotske ruke poslužuje, te pri povezivanju, očitavaju se stanja senzora savitljivosti na rukavici, zatim se očitane vrijednosti formatiraju u JSON, te prije slanja se pretvaraju u niz simbola (*eng. char array*), pri primanju zahtjeva mikroupravljač ESP8266 robotske ruke rasčlanjuje dobivene vrijednosti u *integere*, te *servo* motore postavlja na dobivene vrijednosti, i vraća da je primljena vrijednost mikroupravljaču ESP8266 rukavice.



Sl. 4.5 Rukavica i ruka u krajnjoj ispruženoj poziciji

Na slici 4.5 su prikazane ruka i rukavica. U tom položaju su motori u krajnjem ispruženom položaju prstiju robotske ruke. U ispruženoj poziciji senzori imaju najmanji otpor.



Sl. 4.6 Rukavica i ruka u savinutoj poziciji

Ruka i rukavica su savinute na slici 4.6. Motori su u krajnjoj poziciji radi sigurnosti kako konac ne bi pukao. Prilikom savijanja se otpusti konac *servo* motora za ispravljanje modela ruke (stražnji) i stegne se konac *servo* motora za zatezanje modela ruke (prednji). Savijanjem senzora se otpor u senzoru povećava do  $90^\circ$ , kada je otpor senzora najveći kao što se može vidjeti na slici 4.7.



Sl. 4.7 Otpor na sensorima

## 5. Zaključak

U ovom seminarskom radu napravljena je robotska ruka i upravljačka rukavica koja upravlja robotskom rukavicom. Cilj je bio upravljati robotsku ruku pomoću upravljačke rukavice bežično. Upravljačka rukavica je kreirana mikroupravljačima, sensorima savijanja, i *WiFi* komunikacijom. Model robotske ruke je napravljen 3D printanjem, a sastavljen je vijcima i ljepilom, a za pomicanje prstiju se koristio konac. Za kreiranje ovog rada potrebna su bila znanja iz elektrotehnike, programiranja i mehanike. Ovaj projekt je koncept upravljanja robotskom rukom s upravljačkom rukavicom. Tijekom testiranja bitno je pažljivo postaviti ograničenja gibanja *servo* motora kako ne bi došlo do pucanja kevlarskog konca. To se čini gledanjem krajnje točke pomaka prsta, osjetljivosti gibanja prsta na male promjene zadane veličine, te ukupnim gibanjem prsta ovisno o rotaciji motora. Moguće je napraviti kompleksnija izvedba ruke s više stupnjeva pokreta ili individualnim upravljanjem članaka na prstima, ali za to je potrebno promijeniti model ruke i dodati više komponenti kao što su dodatni motori. Također je moguće dodati preciznije načine detekcije prstiju koristeći potenciometre na svakom individualnom članku.

## LITERATURA

- [1] Gaël Langevin, *InMoov model ruke i podlaktice*,  
<https://inmoov.fr/hand-and-forarm/>, Pristup: 15.8.2023.
- [2] Gaël Langevin, *3D prikaz sastavljanja modela InMoov robotske ruke*,  
<https://inmoov.fr/build-yours/hand-and-forarm-assembly-3d-views/>, Pristup: 17.8.2023.
- [3] *Podatkovna tablica MG996R servo motora*,  
[https://www.electronicoscaldas.com/datasheet/MG996R\\_Tower-Pro.pdf](https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf), Pristup: 17.8.2023.
- [4] *Podatkovna tablica senzora savijanja*,  
<https://components101.com/sensors/flex-sensor-working-circuit-datasheet>, Pristup:  
18.8.2023.
- [5] *Podatkovna tablica ESP8266*,  
[https://soldered.com/productdata/2016/01/Soldered\\_ESP8266-ESP12\\_datasheet.pdf](https://soldered.com/productdata/2016/01/Soldered_ESP8266-ESP12_datasheet.pdf), Pristup:  
20.8.2023.
- [6] *Podatkovna tablica TCA9548A*,  
[https://soldered.com/productdata/2022/03/Soldered\\_tca9548a\\_datasheet.pdf](https://soldered.com/productdata/2022/03/Soldered_tca9548a_datasheet.pdf), Pristup:  
22.8.2023.
- [7] *Podatkovna tablica ADS1015*,  
[https://soldered.com/productdata/2022/03/Soldered\\_ADS1015\\_datasheet.pdf](https://soldered.com/productdata/2022/03/Soldered_ADS1015_datasheet.pdf), Pristup:  
22.8.2023.

## SAŽETAK

Naslov: Robotska ruka upravljana rukavicom

U završnome radu izrađena robotska ruka koja je upravljana bežičnom rukavicom. Rad se sastoji od modela ruke, ESP8266 Wi-Fi modula, *servo* motora, senzora savijanja, *joystick*-a. Tijekom rada objašnjeni su postupak izrade modela robotske ruke, te mehanički principi rada. Također je opisano programsko rješenje upravljanja robotskom rukom pomoću upravljačke rukavice.

**Ključne riječi:** ESP8266, *Wi-Fi*, mikroupravljač, *servo* motor, *flex* senzor, *joystick*

## **ABSTRACT**

**Title:** Robotic arm controlled using a glove

This paper shows construction and working principle of a humanoid robotic arm model. Model contains, ESP8266 Wi-Fi module, servo motor, flex sensor, joystick. Paper also describes the proces of creating model of a robotic arm and describes it mechanically works. Software solution for controlling the robotic arm using a control glove is also described.

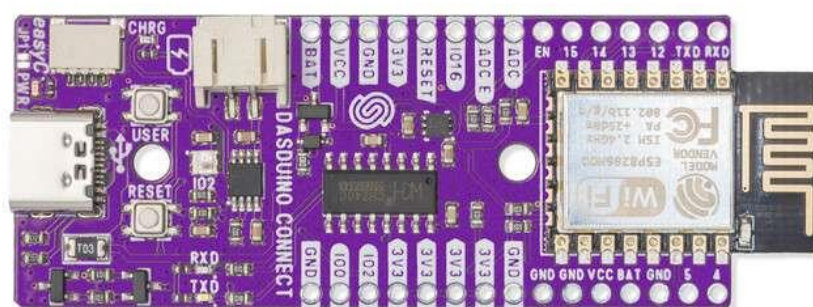
**Key words:** ESP8266, *Wi-Fi*, microcontroller, *servo* motor, *flex* sensor, *joystick*

## **ŽIVOTOPIS**

Kristijan Kesić rođen 15. srpnja.1998 godine u Osijeku. Pohađa Osnovnu školu Ljudevita Gaja u Osijeku od 2005. godine. Svoje obrazovanje nastavlja upisujući Elektrotehničku i prometnu školu Osijek 2013. godine. Upisuje Fakultet elektrotehnike i računarstva i informacijskih tehnologija u Osijeku 2018. godine na stručnom studiju elektrotehnike smjer automatika 2018. godine.

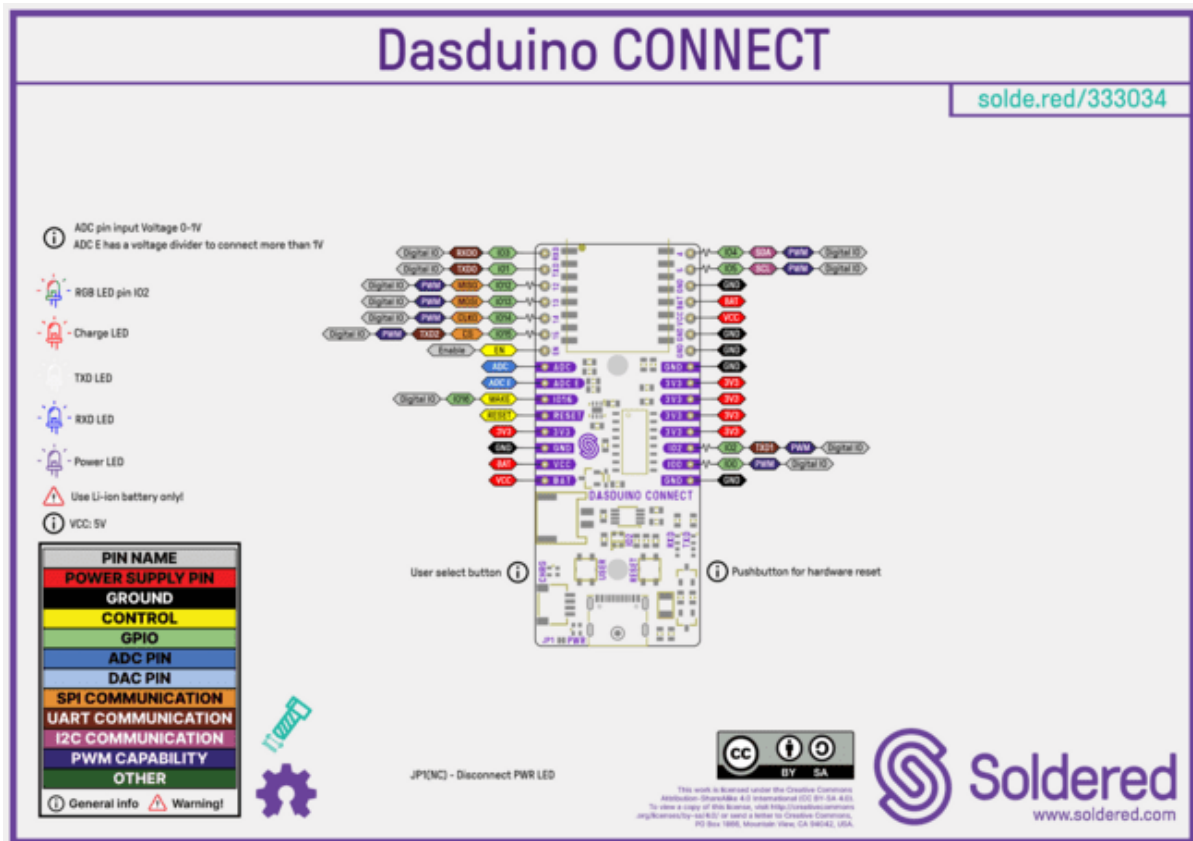
## PRILOZI

[P1] Dasduino CONNECT *WiFi* modul

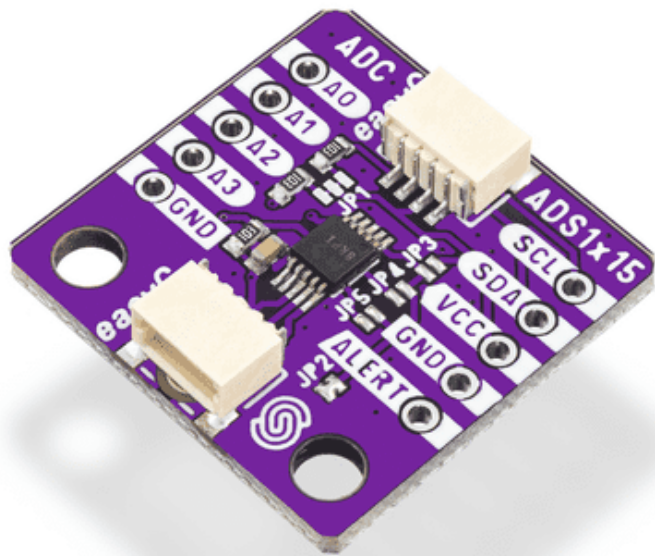




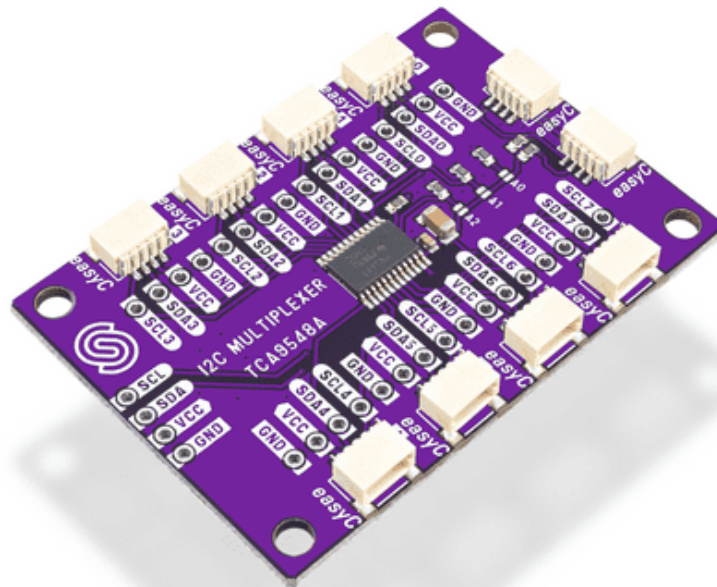
[P2] Raspored pinova na Dasduino CONNECT-u



[P3] ADS1015 multipleksor



[P4] TCA9548A multipleksor



[P3] Servo motor MG996R



[P4] Analogni Joystick



[P5] 3D pišač Creality Ender 3



[P6] Kod upravljačke rukavice ESP8266

```

1 //Inicijalizacija biblioteka
2 #include <Arduino.h>
3 #include <ESP8266WiFi.h>
4 #include <ESP8266HTTPClient.h>
5 #include <WiFiClient.h>
6 #include <ArduinoJson.h>
7 #include <ADS1015-SOLDERED.h>
8 #include <TCA9548A-SOLDERED.h>
9
10 ##Definiranje SSID-a i lozinke za povezivanje na HOTSPOT (Pristupnu Tačka)
11 #ifndef APSSID
12 #define APSSID "ServerRuka"
13 #define APPSK "passwordzaruku"
14 #endif
15
16 const char* SSID = APSSID;
17 const char* PASSWORD = APPSK;
18
19 const char* SERVER_NAME = "http://192.168.4.1:80/setServo";
20
21 //Konstante za Pinove Flex Senzora
22 const u_int8_t ADS1ThumbPin = 0;
23 const u_int8_t ADS1IndexPin = 1;
24 const u_int8_t ADS1MiddlePin = 2;
25 const u_int8_t ADS1RingPin = 3;
26 const u_int8_t ADS2LittlePin = 0;
27 const u_int8_t ADS2WristPin = 1;
28
29 //Konstante za vrijednosti flex senozra i maksimalne kutove servo motora
30 // Palac
31 const u_int16_t THUMB_FLEXED_SENSOR = 696;
32 const u_int16_t THUMB_RELAXED_SENSOR = 53;
33 const u_int8_t MINIMAL_ANGLE_THUMB = 0;
34 const u_int8_t MAXIMUM_ANGLE_THUMB = 130;
35
36 // Kažiprst
37 const u_int16_t INDEX_FLEXED_SENSOR = 1023;
38 const u_int16_t INDEX_RELAXED_SENSOR = 680;
39 const u_int8_t MINIMAL_ANGLE_INDEX = 0;
40 const u_int8_t MAXIMUM_ANGLE_INDEX = 155;
41
42 // Srednjak
43 //Ovaj je naopačke postavljen
44 const u_int16_t MIDDLE_FLEXED_SENSOR = 1043;
45 const u_int16_t MIDDLE_RELAXED_SENSOR = 80;
46 const u_int8_t MINIMAL_ANGLE_MIDDLE = 155;
47 const u_int8_t MAXIMUM_ANGLE_MIDDLE = 0;
48
49 // Prstenjak
50 const u_int16_t RING_FLEXED_SENSOR = 706;
51 const u_int16_t RING_RELAXED_SENSOR = 10;
52 const u_int8_t MINIMAL_ANGLE_RING = 0;
53 const u_int8_t MAXIMUM_ANGLE_RING = 150;
54

```

```

55 // Mali prst
56 const u_int16_t LITTLE_FLEXED_SENSOR = 1036;
57 const u_int16_t LITTLE_RELAXED_SENSOR = 0;
58 const u_int8_t MINIMAL_ANGLE_SMALL = 0;
59 const u_int8_t MAXIMUM_ANGLE_SMALL = 100;
60
61 // Zglob
62 const u_int16_t MINIMAL_WRIST_ANGLE = 0;
63 const u_int16_t MAXIMUM_WRIST_ANGLE = 180;
64
65 //Vrijednosti očitane s flex senzora
66 u_int16_t thumbValue;
67 u_int16_t indexFingerValue;
68 u_int16_t middleFingerValue;
69 u_int16_t ringFingerValue;
70 u_int16_t smallFingerValue;
71 u_int16_t servoWristValue;
72
73 //Timer
74 unsigned long LAST_TIME = 0;
75 // u milisekundama
76 unsigned long TIMER_DELAY = 150;
77
78 // JSON U koji će se spremiti očitane vrijednosti prije slanja na ruku
79 const size_t JSON_BUFFER_SIZE = 256;
80 StaticJsonDocument<JSON_BUFFER_SIZE> doc;
81
82 char json_read_sensors[JSON_BUFFER_SIZE];
83
84 // Inicijalizacija multipleksora i Analogno Digitalnog pretvarača
85 TCA9548A multiplexor;
86 ADS1015 ads1;
87 ADS1015 ads2;
88
89 u_int8_t local_map_function(u_int16_t readValue, u_int16_t relaxedSensor, u_int16_t flexedSensor, u_int8_t minimalAngle, u_int8_t maximumAngle){
90
91     u_int8_t mappedValue = (readValue - relaxedSensor) * ( maximumAngle - minimalAngle ) / ( flexedSensor - relaxedSensor ) + minimalAngle;
92     return mappedValue;
93 }
94
95
96 // Funkcija za čitanje vrijednosti senzora preko analogno digitalnog pretvarača koji je spojen na multipleksor, te ograničavanje vrijednosti u doz
97 u_int8_t read_sensor_from_ads( ADS1015 ads, u_int8_t pinNumber, u_int16_t relaxedSensor, u_int16_t flexedSensor, u_int8_t minimalAngle, u_int8_t m
98
99     u_int16_t sensorReadValue = ads.readADC( pinNumber );
100     if ( sensorReadValue < relaxedSensor ){
101         sensorReadValue = relaxedSensor;
102     } else if ( sensorReadValue > flexedSensor )
103     {
104         sensorReadValue = flexedSensor;
105     }
106

```

```

107     u_int8_t mappedValue;
108     mappedValue = local_map_function(sensorReadValue, relaxedSensor, flexedSensor, minimalAngle, maximumAngle);
109     // mappedValue = sensorReadValue >> 2;
110
111
112     u_int8_t constrainedMappedValue = constrain(mappedValue, minimalAngle, maximumAngle);
113
114     return constrainedMappedValue;
115 }
116
117 // Funkcija za spremanje i pretvaranje očitanih vrijednost u JSON format
118 void read_sensors_to_json(JsonDocument& jsonDoc){
119
120     multiplexor.openChannel(0);
121
122     delay(70);
123
124     thumbValue = read_sensor_from_ads(ads1, ADS1ThumbPin, THUMB_RELAXED_SENSOR, THUMB_FLEXED_SENSOR, MINIMAL_ANGLE_THUMB, MAXIMUM_ANGLE_THUMB);
125
126     indexFingerValue = read_sensor_from_ads(ads1, ADS1IndexPin, INDEX_RELAXED_SENSOR, INDEX_FLEXED_SENSOR, MINIMAL_ANGLE_INDEX, MAXIMUM_ANGLE_INDEX);
127     middleFingerValue = read_sensor_from_ads(ads1, ADS1MiddlePin, MIDDLE_RELAXED_SENSOR, MIDDLE_FLEXED_SENSOR, MINIMAL_ANGLE_MIDDLE, MAXIMUM_ANGLE_MIDDLE);
128     ringFingerValue = read_sensor_from_ads(ads1, ADS1RingPin, RING_RELAXED_SENSOR, RING_FLEXED_SENSOR, MINIMAL_ANGLE_RING, MAXIMUM_ANGLE_RING);
129
130     multiplexor.closeChannel(0);
131     multiplexor.openChannel(1);
132
133     delay(70);
134     smallFingerValue = read_sensor_from_ads(ads2, ADS2LittlePin, LITTLE_RELAXED_SENSOR, LITTLE_FLEXED_SENSOR, MINIMAL_ANGLE_SMALL, MAXIMUM_ANGLE_SMALL);
135
136
137     u_int16_t servoReadValue;
138     servoReadValue = analogRead(17);
139     //u_int8_t mappedWristValue = servoReadValue >> 2;
140     //servoWristValue = constrain( mappedWristValue, MINIMAL_WRIST_ANGLE, MAXIMUM_WRIST_ANGLE );
141     u_int8_t mappedWristValue = local_map_function(servoReadValue, 0, 1023, 0, 180);
142
143     multiplexor.closeChannel(1);
144
145     jsonDoc["thumbValue"] = thumbValue;
146     jsonDoc["indexFingerValue"] = indexFingerValue;
147     jsonDoc["middleFingerValue"] = middleFingerValue;
148     jsonDoc["ringFingerValue"] = ringFingerValue;
149     jsonDoc["smallFingerValue"] = smallFingerValue;
150     jsonDoc["servoWristValue"] = servoWristValue;
151
152
153
154
155     serializeJson(jsonDoc, json_read_sensors);
156     Serial.println(json_read_sensors);
157 }
158
159

```



```

160 void setup() {
161
162
163
164     doc["thumbValue"] = 0;
165     doc["indexFingerValue"] = 0;
166     doc["middleFingerValue"] = 180;
167     doc["ringFingerValue"] = 0;
168     doc["smallFingerValue"] = 0;
169     doc["servoWristValue"] = 90;
170
171     // Početak Serijske komunikacije
172     Serial.begin(115200);
173     Serial.println();
174
175     // Inicijalizacija WiFi-a i povezivanje na zadanu pristupnu točku
176     WiFi.begin(SSID, PASSWORD);
177
178     Serial.print("Connecting");
179     while (WiFi.status() != WL_CONNECTED)
180     {
181         delay(500);
182         Serial.print(".");
183     }
184     Serial.println();
185
186     Serial.print("Connected, IP address: ");
187     Serial.println(WiFi.localIP());
188
189     // Pokretanje multipleksora
190     multiplexor.begin();
191
192     // Pokretanje Analogno digitalnog pretvarača
193     ads1.begin();
194     ads2.begin();
195
196     ads1.setGain(0);
197     ads2.setGain(0);
198
199     ads1.setMode(0);
200     ads2.setMode(0);
201
202     multiplexor.closeAll();
203
204 }
205

```

```

206 void loop() {
207
208     // Uvjet za timer
209     if ((millis() - LAST_TIME) > TIMER_DELAY) {
210         if(WiFi.status()== WL_CONNECTED){
211             WiFiClient client;
212             HTTPClient http;
213
214             // Povezivanje na zadanu adresu
215             http.begin(client, SERVER_NAME);
216
217             // Pozivanje funkcije za očitavanje senzora
218             read_sensors_to_json(doc);
219
220             // Postavljanje header-a zahtjeva
221             http.addHeader("Content-Type", "text/plain");
222             // Data to send with HTTP POST
223             char httpRequestData[JSON_BUFFER_SIZE];
224             // Kopiranje JSON-a očitanih senzora u niz char-a kako bi se mogli poslati na server
225             strcpy(httpRequestData, json_read_sensors);
226             Serial.println(httpRequestData);
227
228             // Slanje HTTP POST-a zahtjeva
229             int httpResponseCode = http.POST(String(httpRequestData));
230
231             Serial.print("HTTP Response code: ");
232             Serial.println(httpResponseCode);
233
234             // Oslobađanje resursa
235             http.end();
236         }
237         else {
238             Serial.println("WiFi Disconnected");
239         }
240         // Resetiranje timer-a
241         LAST_TIME = millis();
242     }
243 }
244

```

[P8] Kod robotske ruke

```

1 //Inicijalizacija biblioteka
2 #include <ESP8266WiFi.h>
3 #include <ESP8266WebServer.h>
4 #include <Servo.h>
5 #include <ArduinoJson.h>
6 #include <Arduino.h>
7
8
9 //Definiranje SSID-a i lozinke za HOTSPOT (Pristupnu Točku)
10 #ifndef APSSID
11 #define APSSID "ServerRuka"
12 #define APPSK "passwordzaruku"
13 #endif
14
15 const char *SSID = APSSID;
16 const char *PASSWORD = APPSK;
17
18 //Konstante za Pinove Servo motoracv
19 const u_int8_t THUMB_SERVO_PIN = 4;
20 const u_int8_t INDEX_SERVO_PIN = 12;
21 const u_int8_t MIDDLE_SERVO_PIN = 13;
22 const u_int8_t RING_SERVO_PIN = 14;
23 const u_int8_t SMALL_SERVO_PIN = 15;
24 const u_int8_t WRIST_SERVO_PIN = 5;
25
26
27 // Postavljanje Static IP adrese kako bi se klijent mogao spojiti i nakon reseta
28 IPAddress apIP(192, 168, 4, 1);
29 IPAddress gateway(192, 168, 1, 1);
30 IPAddress subnet(255, 255, 255, 0);
31
32 //Otvaranje porta 80 za WebServer
33 ESP8266WebServer server(80);
34
35 const int JSON_BUFFER_SIZE = 96;
36
37 //Inicijalizacija Servo-a
38 Servo thumb;
39 Servo indexFinger;
40 Servo middleFinger;
41 Servo ringFinger;
42 Servo smallFinger;
43 Servo wrist;
44
45 //Vrijednosti koje će primiti od rukavice, te se ispisati na servo motore
46 int servoThumbValue = 0;
47 int servoIndexValue = 0;
48 int servoMiddleValue = 180;
49 int servoRingValue = 0;
50 int servoSmallValue = 0;
51 int servoWristValue;
52

```

```

53 //Timer
54 unsigned long LAST_TIME = 0;
55 const long TIMER_DELAY = 90;
56
57 //Pisanje primljenih vrijednosti na Servo Motore
58 void write_servos(){
59     thumb.write(servoThumbValue);
60     indexFinger.write(servoIndexValue);
61     middleFinger.write(servoMiddleValue);
62     ringFinger.write(servoRingValue);
63     smallFinger.write(servoSmallValue);
64     wrist.write(servoWristValue);
65 }
66
67
68 void setup() {
69
70     //Pokretanje Serijske komunikacije
71     Serial.begin(115200);
72
73     // Postavljanje "Soft Access Point-a"
74     WiFi.mode(WIFI_AP);
75     WiFi.softAP(SSID, PASSWORD);
76     WiFi.softAPConfig(apIP, apIP, subnet);
77     //Default 192.168.4.1
78
79     //Ispis SSID-a i Lozinke za Spajanje na WiFi server
80     Serial.print("SoftAP IP Address: ");
81     Serial.println(WiFi.softAPIP());
82
83     Serial.print(F("Creating access point named: "));
84     Serial.print(SSID);
85     Serial.print(F(" and password: "));
86     Serial.println(PASSWORD);
87
88     //End Point na Web serveru za GET zahtjev koji vraća zadani odgovor
89     server.on("/", HTTP_GET, [](){
90         server.send(200, "text/html", "ESP8266 SoftAP !");
91     });
92
93     //End Point na Web serveru za POST request koji prima JSON string koji parsira vrijednosti te ih ispisuje na servo motore
94     server.on("/setServo", HTTP_POST, [](){
95
96         DynamicJsonDocument doc(2048); //Inicijalizacija Json varijable
97
98         DeserializationError error = deserializeJson(doc, server.arg("plain"));
99
100         if (error) {
101             Serial.print(F("deserializeJson() failed: "));
102             Serial.println(error.f_str());
103             return;
104         }
105

```

```

106     servoThumbValue = doc["thumbValue"];
107     servoIndexValue = doc["indexFingerValue"];
108     servoMiddleValue = doc["middleFingerValue"];
109     servoRingValue = doc["ringFingerValue"];
110     servoSmallValue = doc["smallFingerValue"];
111     servoWristValue = doc["servoWristValue"];
112
113     Serial.print("servoThumbValue: ");
114     Serial.println(servoThumbValue);
115     Serial.print("servoIndexValue: ");
116     Serial.println(servoIndexValue);
117     Serial.print("servoMiddleValue: ");
118     Serial.println(servoMiddleValue);
119     Serial.print("servoRingValue: ");
120     Serial.println(servoRingValue);
121     Serial.print("servoSmallValue: ");
122     Serial.println(servoSmallValue);
123     Serial.print("servoWristValue: ");
124     Serial.println(servoWristValue);
125
126     server.send(200, "text/plain", "Servo angle set");
127     });
128
129     server.begin();
130
131     //Inicijalizacija Servo-a
132     thumb.attach(THUMB_SERVO_PIN);
133     indexFinger.attach(INDEX_SERVO_PIN);
134     middleFinger.attach(MIDDLE_SERVO_PIN);
135     ringFinger.attach(RING_SERVO_PIN);
136     smallFinger.attach(SMALL_SERVO_PIN);
137     wrist.attach(WRIST_SERVO_PIN);
138
139 }
140
141 void loop() {
142
143
144     //Pokretanje client handler-a
145     server.handleClient();
146
147     // Timer koji pri istjecanju ispisuje vrijednosti na servo
148     if ((millis() - LAST_TIME) > TIMER_DELAY) {
149
150         write_servos();
151         LAST_TIME = millis(); //reset timer-a
152
153     }
154 }

```