

# Robotska ruka upravljana Arduinom

---

Lapčević, Zoran

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:869167>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-01-13**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**ROBOTSKA RUKA UPRAVLJANA ARDUINOM**

Rijeka, rujan 2023.

Zoran Lapčević

0069080092

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**ROBOTSKA RUKA UPRAVLJANA ARDUINOM**

Mentor: doc. dr. sc. Ivan Volarić

Rijeka, rujan 2023.

Zoran Lapčević

0069080092

Izvornik original zadatka

## Izjava o samostalnoj izradi rada

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija Tehničkog fakulteta, Sveučilišta u Rijeci, izjavljujem da sam isključivi autor rada pod nazivom ROBOTSKA RUKA UPRAVLJANA ARDUINOM uz pomoć mentora doc. dr. sc. Ivan Volarić. U navedenom radu nisu na nedozvoljen način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Rijeka, rujan 2023.

---

Zoran Lapčević

0069080092

# SADRŽAJ

1. UVOD.....	1
2. ROBOTSKA RUKA .....	2
2.1. Podjela robotskih ruku.....	3
2.2. Primjena robotske ruke.....	5
3. KOMPONENTE ROBOTSKE RUKU.....	7
3.1. Arduino Uno .....	7
3.1.1. Komponente Arduino Uno pločice.....	9
3.2. Servomotor .....	11
3.2.1. Pulsno-širinska modulacija.....	12
3.2.2. Upravljanje servo motorom PWM signalom.....	14
3.3. Potenciometar .....	15
3.4. Ostale komponente .....	16
3.4.1. Svjetleća dioda .....	16
3.4.2. Tipkalo.....	18
4. IZRADA ROBOTSKE RUKU.....	19
5. PROGRAMSKI KOD ARDUINO UNO ZA ROBOTSKU RUKU.....	23
6. PRINCIP RADA ROBOTSKE RUKU .....	28
7. ZAKLJUČAK .....	29
Literatura .....	30
Sažetak i ključne riječi .....	32
Summary and key words .....	33

# 1. UVOD

U posljednjim desetljećima, velik napredak u području robotike rezultirao je razvojem mnogobrojnih inovativnih tehnologija. Jedno od najznačajnijih područja istraživanja u robotici je razvoj robotskih ruka, koje predstavljaju ključni dio robota koji omogućuje manipulaciju i interakciju s okolinom. Robotske ruke imaju široku primjenu u različitim industrijama, poput proizvodnje, medicinske dijagnostike, istraživanja svemira i mnogih drugih područja.

U sklopu ovog završnog rada fokus je usmjeren na dizajn, izradu i primjenu robotske ruke koja je upravljana pomoću Arduino razvojne pločice. Uz Arduino u konstrukciju robotske ruke uključeno je i nekoliko servo motora koji su upravljani preko potenciometara koji su spojeni na Arduino. Cilj ovog rada je istražiti mogućnosti izrade robotske ruke upravljane Arduinom te demonstrirati njenu primjenu kroz praktične primjere. Uzimajući u obzir dostupne materijale i resurse, proučeni su različiti koncepti dizajna robotske ruke, kao i razni načini upravljanja i kontroliranja pokreta. Posebna pažnja je usmjerena prema mehanizmima hvatanja objekata i preciznosti pokreta, koji su ključni elementi za uspješno izvođenje manipulacijskih zadataka.

Uz razvoj robotske ruke, istražene su mogućnosti integracije drugih senzora i modula na platformu Arduina kako bi se proširila funkcionalnost i sposobnosti robotske ruke. Osim toga, razmotreni su sigurnosni aspekti vezani uz upravljanje robotskom rukom i načini minimiziranja rizika od ozljeda ili štete.

Kroz praktične primjere, demonstrirano je kako robotska ruka upravljana Arduinom može obavljati različite zadatke, poput pomicanja objekata različitih veličina i oblika, preciznog pozicioniranja ili čak izvođenja jednostavnih manipulacija.

## 2. ROBOTSKA RUKA

Robotske ruke su strojevi koji su programirani da izvršavaju određeni zadatak brzo, učinkovito i izuzetno precizno. Obično su pokretane motorima. Dizajnirane su za brzo, dosljedno izvođenje teških ili lakih ponavljajućih postupaka tijekom dugih razdoblja. Robotske ruke imaju nekoliko različitih komponenata koje rade zajedno kako bi osigurale učinkovito izvršavanje zadatka. Kontroleri su glavni procesori robotskih ruku te oni čine upravljačke konzole koje dolaze u nekoliko oblika ovisno o potrebnoj računalnoj snazi. Mogu biti postavljeni u automatski režim rada ili mogu biti ručno upravljani primanjem naredbi od strane korisnika. Kontrukcija robotske ruke, prikazana na slici 2.1., je također veoma važna. Sastoji se od tri glavna dijela: ramena, lakta i zgloba. Rame se nalazi na bazi ruke te su na njemu pričvršćeni kontroleri, te se rame može rotacijski gibati oko dvije osi. Lakat se nalazi na sredini robotske ruke i preko kojeg se gornji dio ruke rotacijski giba neovisno o donjem dijelu. Na samom kraju robotske ruke se nalazi zglob na koji je povezan krajnji efektor. Krajnji efektor je uređaj koji je dizajniran za interakciju s okolinom. Dizajn krajnjeg efektora može biti različit ovisno o namjeni robotske ruke. Može biti nastavak u obliku hvataljke, a može biti aparat za zavarivanje, bušilica i slično. Sve ovo komponente su pokretane s pogonskim mehanizmima to jest s motorima koji reguliraju kretanje i manevriranje između zglobova. Najšće su to servo motori. Naprednije robotske ruke sadrže i senzore koji omogućuju detektiranje i reagiranje na okolinu.



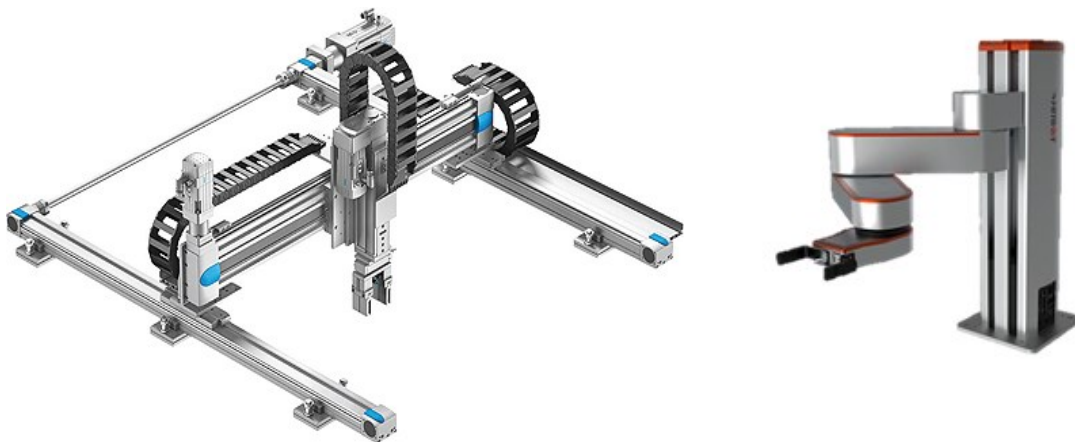
*Slika 2.1. Primjer konstrukcije robotske ruke [12].*



## 2.1. Podjela robotskih ruku

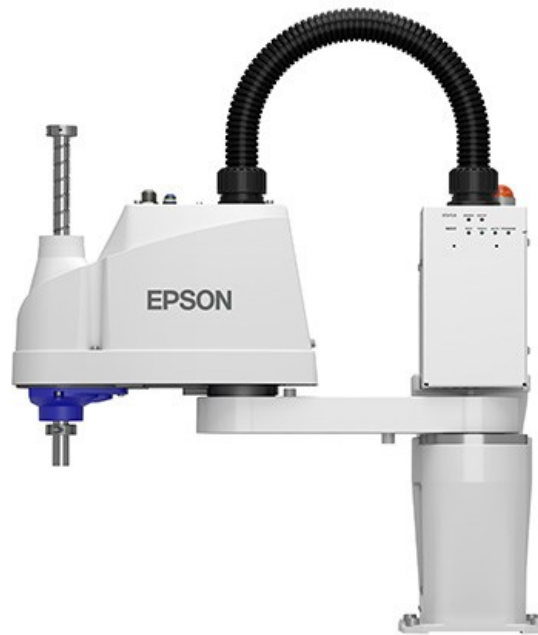
U nastavku je opisana osnovna podjela robotskih ruku.

- a) Zglobna robotska ruka: Zglobna ruka robota, prikazana na slici 2.1., nalikuje ljudskoj ruci i omogućuje mehaničko kretanje i konfiguraciju. To je jedan od najčešćih tipova robotskih ruku za industrijsku automatizaciju. Sadrži jednu mehaničku ruku pričvršćenu na podlogu sa zakretnim zglobom.
- b) Kartezijska robotska ruka: Kartezijske robotske ruke poznate su i kao linearni roboti ili portalni roboti. Ovo su tipovi industrijskih robota koji rade na tri linearne osi koristeći kartezijski koordinatni sustav (X, Y i Z). To znači da se kreću u ravnim linijama po tri različite osi (gore i dolje, unutra i van te sa strane i na stranu). Zbog toga je ova vrsta robotske ruke, prikazana na slici 2.2.(a), korištena najviše u industrijske svrhe koji traže veću fleksibilnost u svojim konfiguracijama.
- c) Cilindrična robotska ruka: Cilindrične robotske ruke dizajnirane sa mogućnosti kretanja vertikalno gore dolje. Imaju rotacijski zglob u bazi i prizmatički zglob koji povezuje dijelove robotske ruke. Ovaj tip robotske ruke, prikazana na slici 2.2.(b), postiže se s pomoću rotirajuće osovine i produživog kraka koji omogućuje vertikalno i klizno kretanje. Pogodni su za male i jednostavne zadatke poput montaže, rukovanja strojevima ili aplikacija za premazivanje.



Slika 2.2. (a) Kartezijska robotska ruka [14], (b) Cilindrična robotska ruka [15].

- d) Delta robotska ruka: Delta robotske ruke također se nazivaju "paralelne robotske ruke" jer omogućavaju tri ruke spojene na jednu bazu postavljenu iznad radnog prostora. Ove vrste robotskih ruku imaju opcije velike brzine i stoga se koriste za automatizaciju.
- e) Polarna (sferna) robotska ruka: Polarne robotske ruke imaju bazu spojene s rukom s dva rotacijska zgloba i jednim linearnim zglobom. Robotske osi kombinirane su na način da tvore polarnu koordinatu. Polarne robotske ruke namijenjene su za lijevanje pod pritiskom, injekcijsko prešanje, zavarivanje i rukovanje materijalima.
- f) SCARA robotska ruka: Selektivna Konformna Montažna Robotska Ruka ,prikazana na slici 2.3., (eng. *Selective Compliance Assembly Robotic Arm*) dizajnirana je za troosno (X, Y i Z) kretanje kombinirano s rotacijskim pokretom. U usporedbi s kartezijskim robotima, SCARA roboti bolje obavljaju lateralna kretanja, brže se kreću i lakše se integriraju. SCARA roboti obično se koriste u biomedicinskim aplikacijama, za paletizaciju i montažu.



Slika 2.3. SCARA robotska ruka [16].

## 2.2. Primjena robotske ruke

Robotske ruke imaju široku primjenu u industriji zbog svoje sposobnosti izvođenja preciznih i ponavljajućih zadataka. Mogu se vidjeti u mnogim sektorima, uključujući proizvodnju, automobilsku industriju, poljoprivredu i medicinu. Robotske ruke omogućuju automatizaciju različitih zadataka i pridonose povećanju učinkovitosti i kvalitete u različitim industrijskim okruženjima.

U proizvodnji i montaži, robotske ruke su ključne za postavljanje komponenata, zavarivanje dijelova i izvođenje preciznih operacija u proizvodnom procesu. Također, u logističkim centrima, koriste se za brzo rukovanje paketima i automatiziranu isporuku. U medicini, robotske ruke omogućuju kirurške zahvate s visokom preciznošću i minimalnom invazijom. Također se koriste u elektroničkoj industriji za postavljanje komponenata na ploče i druge zadatke.

U industriji automobila, prikazano na slici 2.4., su glavni alat za proizvodnju, obavljaju zadatke poput zavarivanja, prikazano na slici 2.6., i montaže vozila. U istraživanju i razvoju, robotske ruke su nezamjenjive za testiranje prototipova i izvođenje preciznih mjerenja. Također se koriste za rukovanje opasnim tvarima ili u zahtjevnim okruženjima gdje ljudi ne bi mogli sigurno djelovati.

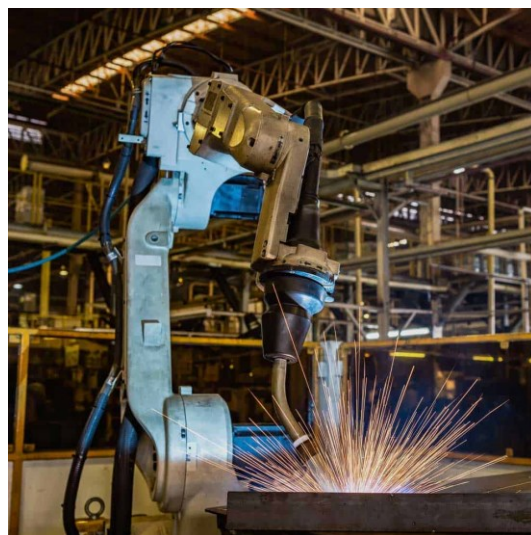


*Slika 2.4. Primjer robotske ruke u automobilskoj industriji [6].*

Robotske ruke predstavljaju veliki napredak u širokom spektru industrijskih sektora te pružaju veliki niz prednosti i unapređenja u proizvodnim procesima. Najveća prednost koju donose robotske ruke je povećanje sigurnosti. One su sposobne izvršavati zadatke u okruženjima koja su opasna za ljude, kao što su toksične kemikalije, visoke temperature ili mjesta s visokim rizikom od ozljeda. Time se smanjuje potencijalna izloženost radnika opasnostima i povećava njihova sigurnost. Također je povećana učinkovitost i produktivnost u proizvodnji. Robotske ruke mogu raditi neprestano i održavati konstantnu produktivnost je izazovnije za ljudski rad. Osim kontantnog rada robotske ruke odlikuje iznimna preciznost u izvršavanju zadataka. To je veoma bitna značajka kada se koriste u medicini ili prilikom montaža preciznih komponenti ili izvođenja točnih mjerenja. Još jedna od ključnih prednosti robotskih ruku je njihova prilagodljivost za podršku raznim aplikacijama, od najjednostavnijih do najkompleksnijih poslova, prikazano na slici 2.5., u sigurnim ili zahtjevnim okruženjima.



*Slika 2.5. Primjer robotske ruke u medicini[8].*



*Slika 2.6. Primjer robotske ruke za zavarivanje [9].*

### 3. KOMPONENTE ROBOTSKE RUKE

#### 3.1. Arduino Uno

Arduino Uno je razvojna pločica baziran na Atmel-ovom mikrokontroleru ATmega328P. Ova razvojna pločica je popularna među entuzijastima, inženjerima i hobistima zbog svoje jednostavnosti korištenja i fleksibilnosti. Jedan od ključnih aspekata Arduina je otvorenost, što omogućava korisnicima da prilagode i programiraju uređaj prema svojim potrebama.

Arduino Uno se ističe svojom kompaktnom veličinom i jednostavnošću sučelja. Jedan od ključnih aspekata Arduina je njegova sposobnost jednostavnog programiranja. Arduino IDE (eng. *Integrated Development Environment*) je besplatna platforma koja omogućava korisnicima pisanje i prenošenje programskog koda na Arduino Uno razvojnu pločicu. Korisnici mogu koristiti jezik sličan C/C++ kako bi kontrolirali uređaje spojene na Arduino, kao što su LED diode, motori, senzori i drugi uređaji.

Središnji element Arduino Uno razvojne pločice je mikrokontroler, konkretno model ATmega328P razvijen od strane *Microchip Technology*. Ovaj mikrokontroler je srce cijelog sustava. On pohranjuje napisani programski kod i obavlja sve računalne operacije potrebne za izvršenje tog koda. S 32 kB flash memorije za programski kod, 2 kB RAM-a za privremene podatke i 1 kB EEPROM-a za trajno pohranjivanje podataka, ATmega328P pruža potrebne resurse za mnoge projekte.

Arduino Uno ima 14 digitalnih ulaza/izlaza koji se mogu koristiti za komunikaciju s vanjskim uređajima. Ovi ulazi/izlazi omogućavaju čitanje i slanje digitalnih signala. Također, šest od ovih pinova omogućavaju pulsno-širinsku modulaciju (eng. *Pulse Width Modulation*) to jest omogućavaju regulaciju srednje vrijednosti napona ili generiranje upravljačkih signala (npr. za servo motore, intezitet svjetline LED dioda, itd.). Pločica također ima šest analognih ulaza (A0-A5) koji omogućavaju mjerenje analognih naponskih vrijednosti.

Pločica se može napajati putem USB priključka, što omogućava programiranje i napajanje iz računala. Također, postoji mogućnost napajanja putem vanjskog izvora. Pločica sadrži i USB priključak za povezivanje s računalom, kristalni oscilator za generiranje radnog takta, reset tipku za ponovno pokretanje programa te LED indikatore za prikazivanje statusa.

Arduino Uno razvojna pločica, prikazana na slici 3.1., ima posebne pinove koji omogućavaju fizičko povezivanje s vanjskim uređajima poput senzora, motora ili drugih modula. Ova pločica je dizajnirana da bude prijateljska za korištenje s proto pločicom, što omogućava korisnicima da lako eksperimentiraju s prototipiranjem i povezivanjem komponenata.

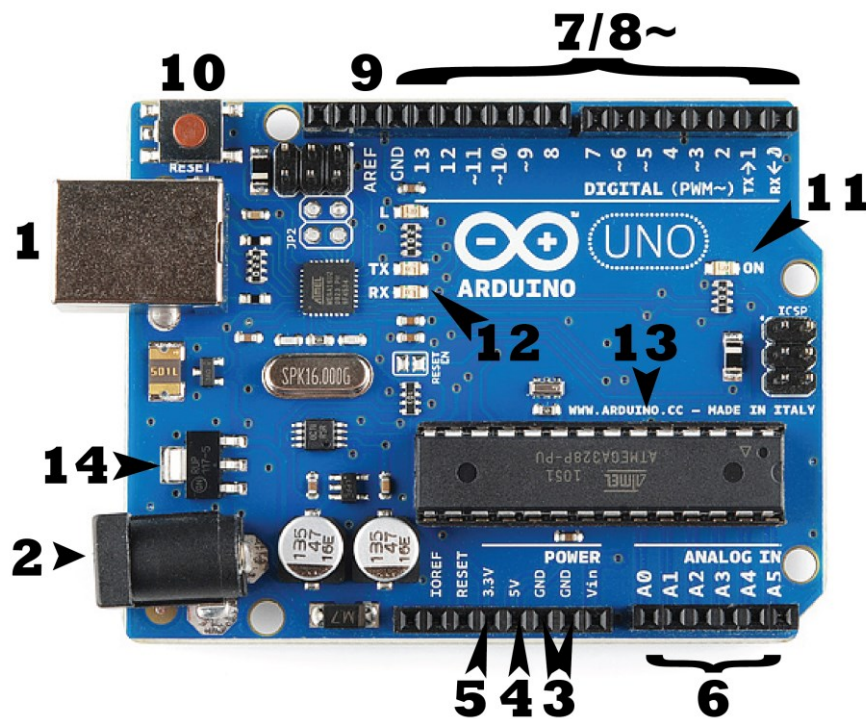


*Slika 3.1. Prikaz Arduino pločice[1].*

Arduino Uno je pogodan za različite projekte, od malih prototipova do složenih sustava. Ova pločica kombinira fizičku platformu i razvojno okruženje kako bi omogućila korisnicima dizajn različitih uređaja, od osnovnih svjetlosnih efekata do složenih interaktivnih sustava. Zahvaljujući širokoj bazi korisnika i zajednici, postoje mnoge biblioteke i resursi koji olakšavaju razvoj projekata. Osim toga, Arduino Uno je pristupačan po cijeni, što ga čini idealnim izborom za početnike koji se upuštaju u svijet mikrokontrolera i elektronike.

### 3.1.1. Komponente Arduino Uno pločice

Arduino Uno razvojna pločica se može napajati na dva načina. Prvi način je preko USB konektora označen s (1) na koji dolazi USB kabel koji je priključen na računalo, a drugi je preko nekog vanjskog izvora napajanja preko priključka za vanjsko napajanje koji je označen s (2). Korištenjem pinova označenih brojevima od (6) do (8), prikazani na slici 3.2., omogućeno je ostvarivanje veze s vanjskim komponentama i sklopovima. U nastavku su kratko opisane funkcionalnosti tih pinova.



Slika 3.2. Prikaz Arduino pločice s brojnim oznakama komponenata [1].

- GND (3): Na Arduino pločici postoji nekoliko GND (ground) pinova, od kojih svaki može poslužiti kao veza prema masi.
- 5V (4) i 3.3V (5): Dva naponska izlaza koja služe za napajanje vanjskog kruga.
- Analogni pinovi (6): Analogni izlazi koji imaju oznake od A0 do A5. Pinovi su povezani s 10-bitnim analognim-digitalnim pretvornikom koji omogućuje čitanje vrijednosti analognog signala.

- d) Digitalni pinovi (7): Digitalni pinovi su oni koji imaju oznake od 0 do 13 na Arduino pločici. Ovi pinovi se mogu koristiti kao digitalni ulazi i kao digitalni izlazi.
- e) PWM (8): Uz neke od digitalnih pinova (3, 6, 9, 10 i 11) se nalazi oznaka (~). Oni se mogu koristiti za pulsno-širinsku modulaciju (*Pulse-Width modulation*). Na taj način na ovim pinovima je moguće generirati analogni signal.
- g) AREF (9): Analogna referenca može se koristiti za pružanje vanjskog referentnog napona za analognu-digitalnu konverziju ulaza na analognim pinovima. Referentni napon je obično jednak napon napajanja analogno-digitalnog pretvornika. Referentni napon suštinski određuje maksimum ulaznog raspona, i posljedično svaki diskretni korak u digitalnom izlazu.
- h) Reset tipkalo (10): Kada se pritisne, taj pin trenutno povezuje pin za resetiranje na logičku nulu, što rezultira resetiranjem mikrokontrolera.
- i) LED indikator napajanja (11): Prilikom priključenja Arduino pločice na izvor napajanja upaliti će se LED dioda.
- j) TX i RX LED (12): Ove LED diode omogućavaju praćenje serijske komunikacije i očitavanje aktivnosti serijskog porta u stvarnom vremenu. TX (eng. *Transmit*) LED se pali svaki put kada mikrokontroler šalje podatke prema vanjskom uređaju ili računalu. RX (eng. *Receive*) LED se pali svaki put kada mikrokontroler prima podatke od vanjskog uređaja ili računala.
- k) Mikrokontroler (13): Mikrokontroler je srce Arduino Uno pločice i omogućava programiranje i kontrolu različitih uređaja i funkcija na pločici. Mikrokontroler u Arduino Uno pločici je ATmega328.
- l) Regulator napona (14): Smanjuje ulazni napon na stabiliziranih (5V) koji su predviđeni za normalan rad mikrokontrolera.



### 3.2. Servomotor

Servo motor, prikazan na slici 3.3., je vrsta električnog motora koji se koristi za precizno upravljanje položajem, brzinom i ubrzanjem rotirajućeg ili linearnog mehanizma. Ovi motori često nalaze primjenu u različitim područjima kao što su industrijski roboti, RC modeli, CNC strojevi, kamere za praćenje pokreta te mnoge druge situacije gdje je ključna visoka preciznost i kontrola pokreta.

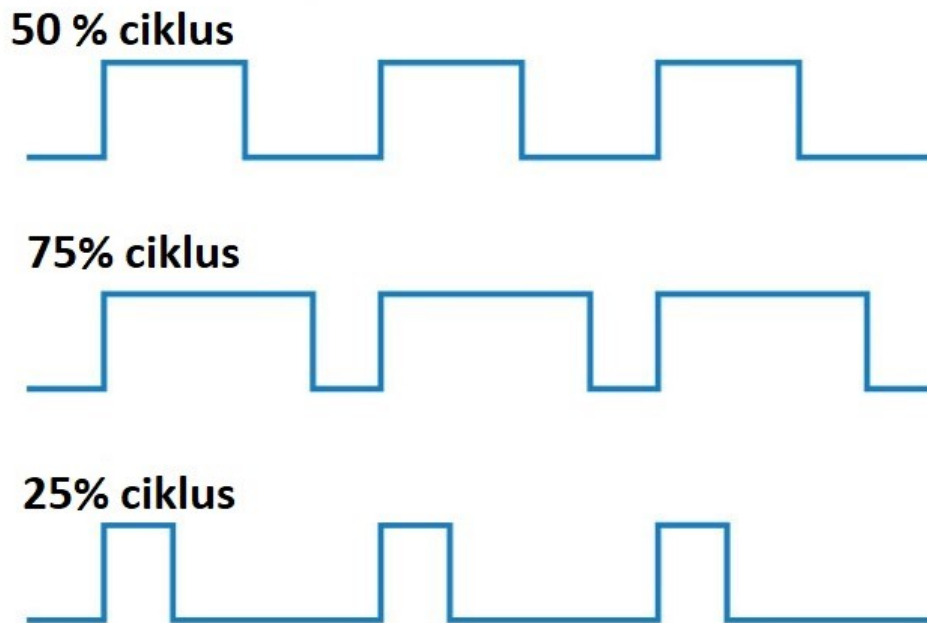


*Slika 3.3. Prikaz servomotora [17].*

Osnovna komponenta servomotora je sam motor, koji može biti električni, hidraulični ili pneumatski, s tim da su električni servomotori najčešći. Uz motor, servomotori često uključuju i senzore povratne informacije kao što su enkoderi ili povratne opruge koje omogućuju precizno praćenje položaja osovine. Jedna od ključnih karakteristika servomotora je mogućnost povratne kontrole. To znači da se pomoću senzora stalno prati stvarni položaj ili brzina osovine i uspoređuje s željenim vrijednostima. Na temelju tog uspoređivanja, upravljački uređaj, često nazivan servo kontrolerom, generira odgovarajući kontrolni signal koji regulira napajanje motora. Ovaj proces se često odvija vrlo brzo, omogućavajući visoku točnost i preciznost u postizanju željenih pokreta. Servo motori se često primjenjuju tamo gdje su potrebni ponovljivi i precizni pokreti. Na primjer, u industrijskim robotima, servomotori omogućuju robotima da precizno pozicioniraju alate ili predmete s velikom preciznošću.

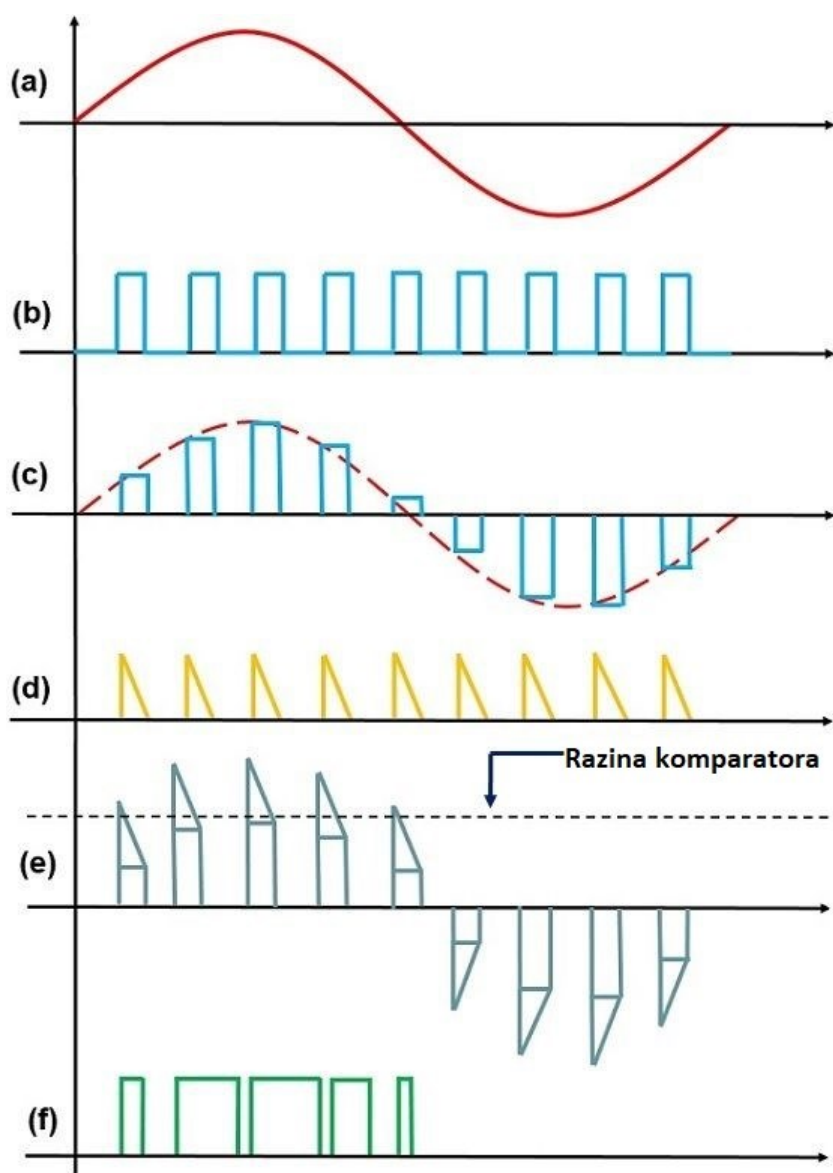
### 3.2.1. Pulsno-širinska modulacija

Pulsno-širinska modulacija definirana je kao modulacijska tehnika u kojoj se širina impulsa pulsnog nositelja mijenja prema modulacijskom signalu. PWM (*Pulse Width Modulation*) je tehnika za kontrolu analognih signala putem digitalnih uređaja. Kvadratni signal je generiran mjenjanjem vremenskog omjera između aktivnog i neaktivnog stanja signala, pri čemu je aktivno stanje obično visoki napon ili logička jedinica, dok je neaktivno stanje obično niski napon ili logička nula. Početni korak u generiranju PWM signala je digitalni izvor, kao što je mikrokontroler. U mikrokontroleru se generira PWM signal uporedbom internog radnog takta s referentnim signalom (modulacijski signal). Referentni signal je analogni signal čijom se vrijednosti kontrolira. Nakon toga dolazi do usporedbe radnog takta s referentnim signalom te se ovisno o usporedbi generira PWM signal. Ako je referentni signal veći od radnog takta, PWM signal će imati dužu aktivnu fazu dok će neaktivna faza biti kraća i obrnuto. PWM signal se sastoji od niza impulse te ukupna duljina impulsa određuje period PWM signala. Frekvencija PWM signala određuje koliko često se ciklička promjena događa u sekundi. Na primjer, za PWM signal, prikazan na slici 3.4., s ciklusom od 50%, signal je u aktivnom stanju (visoki napon) 50% vremena, a u neaktivnom stanju (niski napon) preostalih 50% vremena.



Slika 3.4. Primjer PWM signala [18].

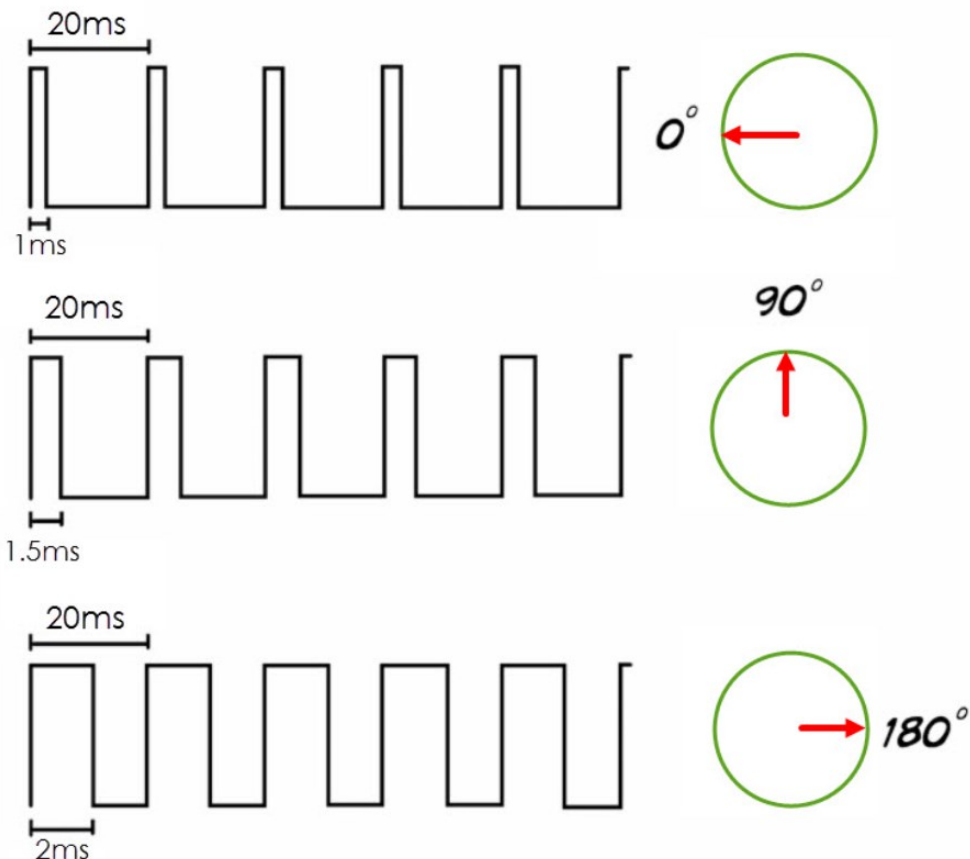
U nastavku je opisan postupak generiranja PWM signala, prikazan na slici 3.5. Prvi signal (a) prikazuje oblik vala sinusoidalnog modulacijskog signala, a drugi signal (b), prikazuje pulsirani nositelj. Nakon modulacije, generira se PAM (*eng. Pulse Amplitude Modulation*) signal koji je prikazan na slici (c). Modulacija amplitude impulsa je modulacijska tehnika koja mijenja amplitudu pulsnog signala kako bi prenijela informacije. Kod sinusoidalnog modulacijskog signala, PAM će prilagoditi visinu impulsa prema vrijednostima amplitude modulacijskog signala u trenutku impulsa. PAM signal zbrojen sa signalom rampe (d) uspoređuje se s referentnim naponom komparatora (e). Na ovaj način je generiran PWM signal (f).



Slika 3.5. Prikaz generiranja PWM signala [19].

### 3.2.2. Upravljanje servo motorom PWM signalom

PWM period koji je specificiran, prikazan na slici 3.6., za *Tower Pro* servo motor koji je korišten prilikom izrade robotske ruke u ovom radu je 20 milisekundi. Za rotaciju servo motora potrebno je generirati PWM signal s periodom od 20 milisekundi i signalnom širinom impulsa između nula do dve milisekunde.

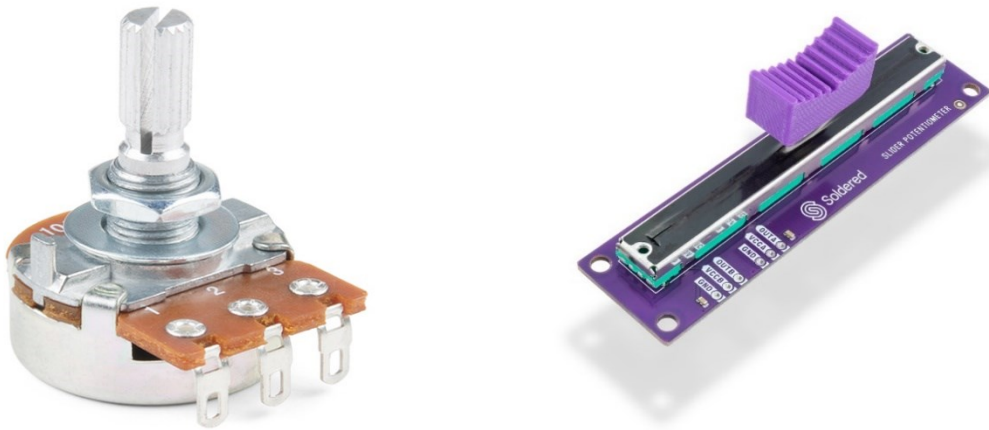


Slika 3.6. PWM signal za *Tower Pro* servo motor [20].

Servo motor je pozicioniran na 90 stupnjeva pri širini impulsa od 1.5 ms. Za okretanje na 180 stupnjeva potrebna je širina impulsa od 2 ms. Prilikom okretanja na 0 stupnjeva potrebna je širina impulsa od 1 ms. Jednostavnom varijacijom širine impulse na 1.75 ms servo se može pomaknuti na kut od 135 stupnjeva. Brzina servo motora se može kontrolirati mijenjanjem širine impulsa vrlo polako. Ako širina impulsa premaši 2 ms, servo motor se neće vratiti u svoj položaj i može doći do oštećenja. Kod nekih servo motora, ako je širina impulsa veća od maksimalne granice, neće utjecati na servo motor tj. neće se rotirati.

### 3.3. Potenciometar

Potenciometar je u ovom završnom radu korišten kao naponsko djelilo. Potenciometar je elektronička komponenta koja se koristi za promjenu električnog otpora u skladu s promjenom mehaničkog položaja pomičnog elementa. Sastoji se od trake otpornog materijala, često karbonske paste ili žice, na kojoj se pomiče kontakt koji ima električnu vezu s terminalima potencijometra. Ovisno o vrsti potencijometra, kontakt se može pomicati rotiranjem oko osovine (rotacijski potencijometar) ili klizanjem po traci (linearni potencijometar), prikazani na slici 3.7.

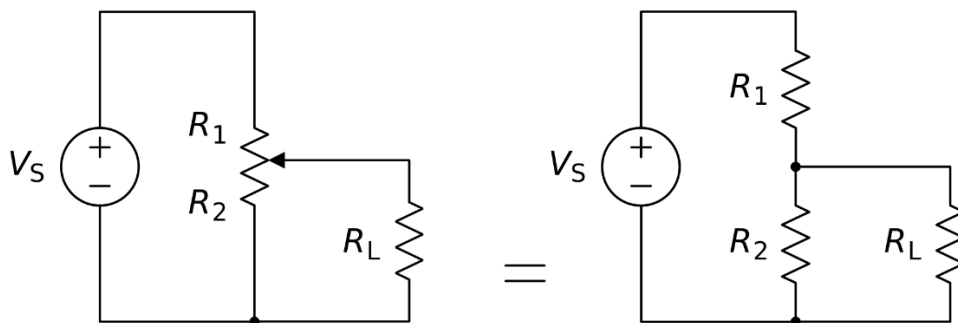


Slika 3.7. Potenciometar: (a) rotacijski, (b) linearni.

Kretanjem kontakta mijenja se dužina otpornog puta, što rezultira promjenom ukupnog otpora između krajnjih kontakata i klizača. Iz slike 3.8., možemo izraziti i omjer ulaznog i izlaznog napona kao:

$$\frac{U_2}{U_1} = \frac{R_2}{R_1 + R_2} \quad , \quad (3.1)$$

gdje je:  $U_2$  izlazni napon,  $U_1$  ulazni napon,  $R_1$  i  $R_2$  početne i krajnje vrijednosti otpora.



Slika 3.8. Shema otpora potencijometra[3].

Potenciometri se koriste u mnogim elektroničkim uređajima za podešavanje različitih parametara kao što su glasnoća zvuka, svjetlina zaslona, frekvencija i ton u audio uređajima te mnoge druge funkcije koje se mogu prilagoditi. Uglavnom se koriste kao analogni uređaji za fino podešavanje ili kontrolu u elektroničkim krugovima.

### 3.4. Ostale komponente

#### 3.4.1. Svjetleća dioda

Svjetleća dioda ili LED (*eng. Light Emitting Diode*) je poluvodička elektronička komponenta, prikazana na slici 3.9., koja ima sposobnost transformacije električnog signala u optički, odnosno svjetlosni signal. Kada je propusno polarizirana, LED dioda emitira elektromagnetsko zračenje zbog procesa poznatog kao elektroluminiscencija. Elektroluminiscencija se temelji na svojstvu poluvodičkih materijala da emitiraju fotone kada elektroni prelaze s viših energetske stanja na niža, što rezultira spontanom oslobađanjem energije u obliku svjetlosti.

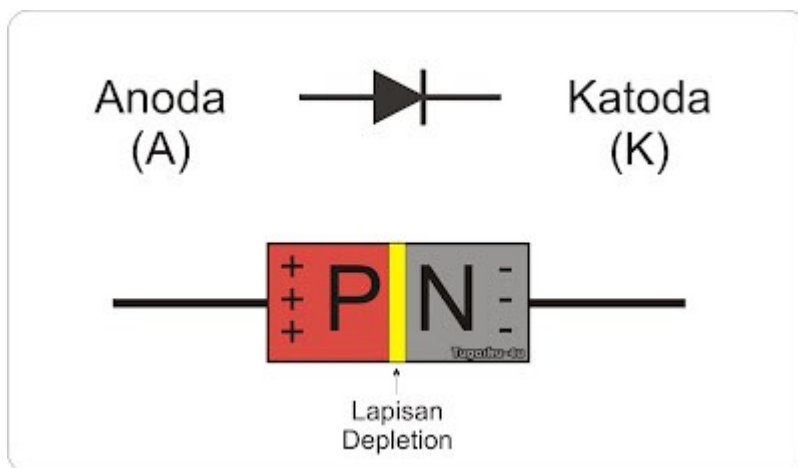


*Slika 3.9. Svjetleća dioda [21].*

U osnovi, LED dioda se sastoji od slojeva različitih poluvodičkih materijala i temelji se na pn-spoju. P-tip spoja obiluje pozitivno nabijenim nositeljima naboja (pozitivni elektroni), a drugi dio spoja n-tip ima više negativno nabijenih nositelja naboja (negativni elektroni). Kada se spoje, dolazi do difuzije nositelja naboja i stvaranja osiromašenog područja između ova dva dijela, gdje su nositelji naboja rijetki. LED diode se često izrađuju od materijala poput galija, arsena i fosfora (GaAsP), pri čemu se omjer tih elemenata kontrolira kako bi se odredila širina zabranjenog pojasa, a time i frekvencija emitiranog zračenja, tj. boja LED diode. Također, postoje i heterostrukturalne izvedbe koje su prikladne za optičke komunikacije, a izrađuju se od istih elemenata uz dodatak indija (In).

Kada je LED dioda propusno polarizirana elektroni se počnu kretati kroz pn spoj. Elektroni s n-tipa prelaze u p-tip, a elektroni s p-tipa prelaze u n-tip. Kada elektroni s viših energetske razine prelaze na niže, dolazi do otpuštanja energije u obliku fotona, što rezultira emisijom svjetlosti.

Arhitektura pn spoja, prikazana na slici 3.10., je ključna za boju svjetlosti koju LED dioda emitira. Različiti materijali i postupci proizvodnje omogućuju proizvodnju LED dioda u različitim bojama, uključujući crvenu, zelenu, plavu i druge. Kombinacijom ovih boja mogu se stvarati bijela svjetlost ili razne nijanse.



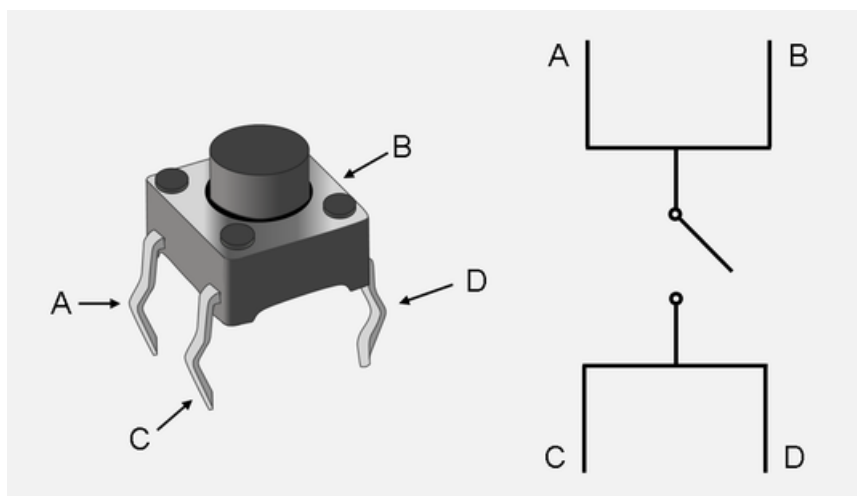
*Slika 3.10. Arhitektura pn spoja (eng. Lapisan Depletion – osiromašeno područje)[9].*

Važno je napomenuti da LED diode karakterizira visoka energetska učinkovitost jer se većina električne energije pretvara u svjetlost, a vrlo mala količina se gubi kao toplina. Također, LED diode imaju dug životni vijek u usporedbi s tradicionalnim izvorima svjetlosti poput žarulja sa žarnom niti.

Zbog svojih brojnih prednosti, LED diode su postale izuzetno popularne i široko korištene u različitim aplikacijama, uključujući rasvjetu u domovima, industrijskim postrojenjima, prometnim signalima, zaslonima na elektroničkim uređajima, medicinskim uređajima i mnogim drugim područjima.

### 3.4.2. Tipkalo

Tipkalo je elektronička komponenta koja omogućava otvaranje ili zatvaranje električnog kruga kada se pritisne. Iako postoje različite vrste tipkala, mehanička tipkala su najčešće korištena u elektronici. Ona se sastoje od dva para pinova te stvaraju kontakt kada se pritisnu, prikazano na slici 3.11. To znači da se zapravo koriste samo dva različita pina koji se inače ne dodiruju. Dok je tipkalo nepritisnuto, ova dva pina nisu u kontaktu.



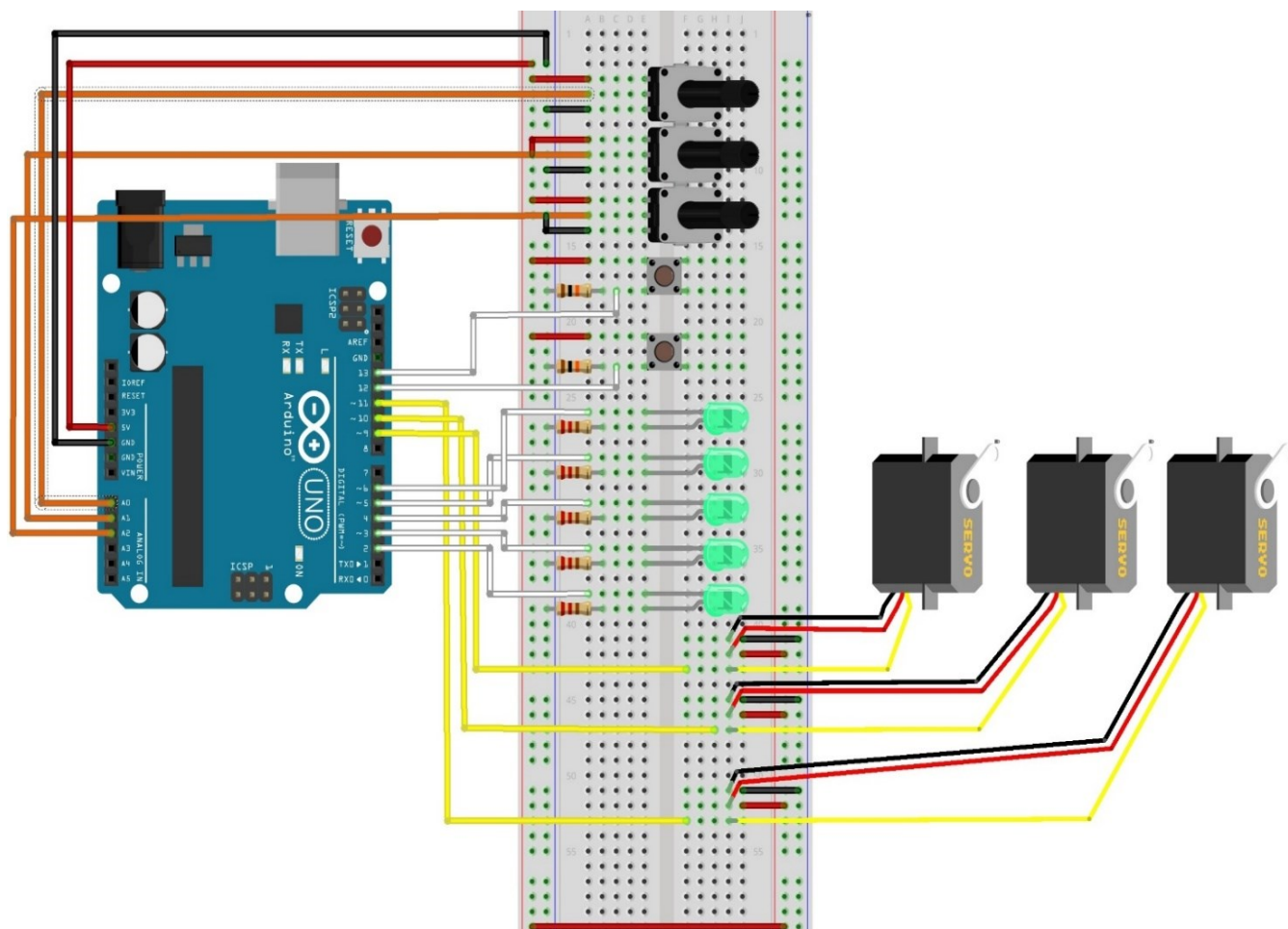
Slika 3.11. Kontrukcija tipkala [11].

Kada se tipkalo pritisne, dolazi do fizičkog dodira između ta dva pina, što zatvara električni krug i omogućava protok struje. Ovaj zatvoreni krug mijenja električno stanje tipkala. Mikrokontroler može detektirati ovo promijenjeno stanje te reagirati na pritisak tipke. Tokom rada tipkala postoji mogućnost dolaska do brzih fluktuacija u električnom signalu zbog fizičkih kontakata koji se odbijaju jedan od drugog prije nego što se stabiliziraju. Ta pojava može uzrokovati da mikrokontroler registrira višestruke pritiske na tipkalu za jedan fizički pritisak. *Debouncing* tipkala je postupak koji rješava ovaj problem te osigurava da se registrira samo jedan signal čak i kad postoje brze promjene. Postupak može biti softverski ili hardverski. Hardverski postupak je implementiranje *debouncing* sklopa koji se sastoji od kombinacije kondenzatora i otpornika. Implementiranjem ovog sklopa uvodi se kratko kašnjenje signala te se tako omogućava stabilizacija bilo kakvih brzih signala. Softverski postupak se izvršava funkcijom *delay* koja izvodi kašnjenje signala u trajanju koje je unaprijed napisano u programskom kodu.



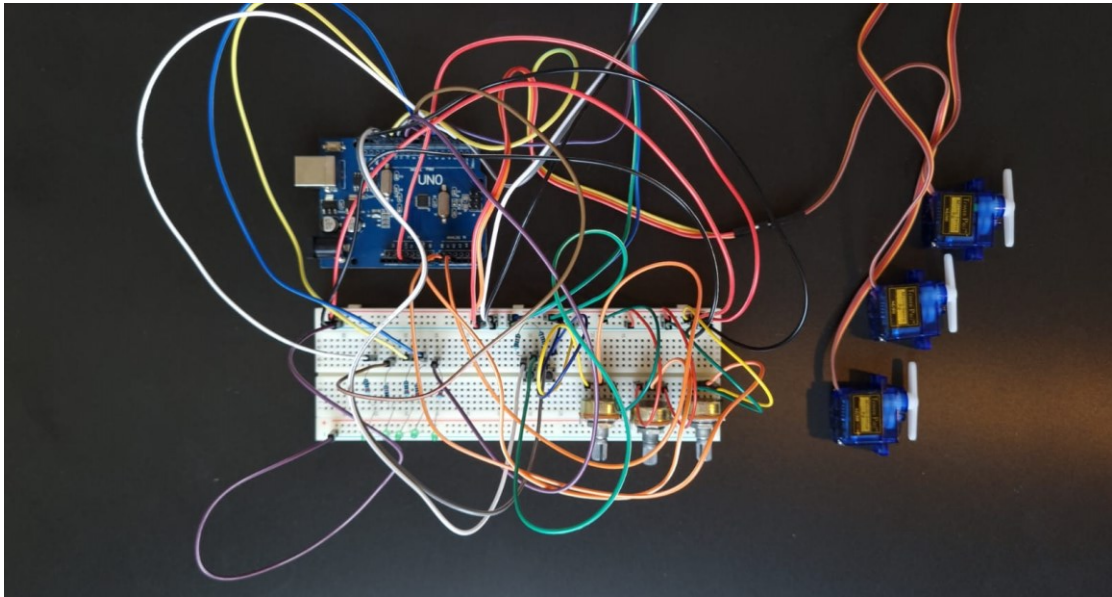
## 4. IZRADA ROBOTSKE RUKE

Izgradnja robotske ruke je započela spajanjem svih elektroničkih komponenata prema shemi prikazanoj na slici 4.1.

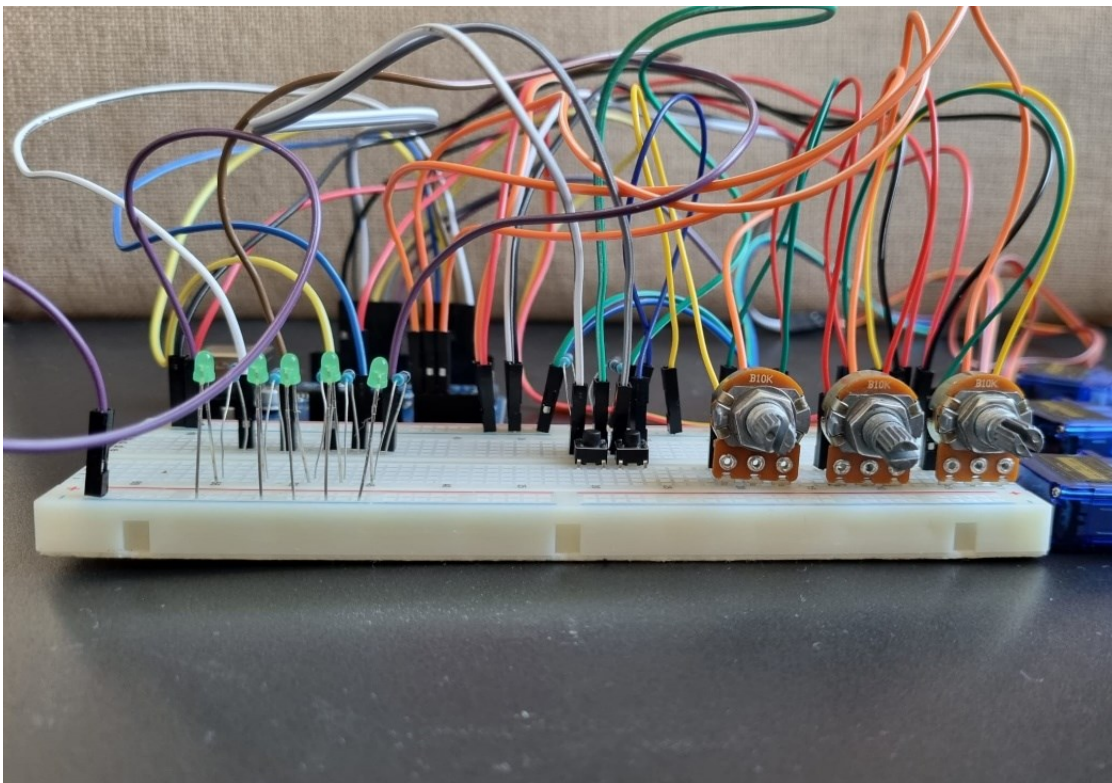


Slika 4.1. Shema spoja u Fritzingu.

Robotska ruka se sastoji od tri servo motora, tri potenciometra, pet otpornika od 220 ohma povezanih s pet dioda, te još dva otpornika od 10k ohma koja su spojena na dva tipkala. Sve ove komponente, prikazane na slikama 4.2. i 4.3., su povezane sa Arduino Uno pločicom i eksperimentalnom pločicom.

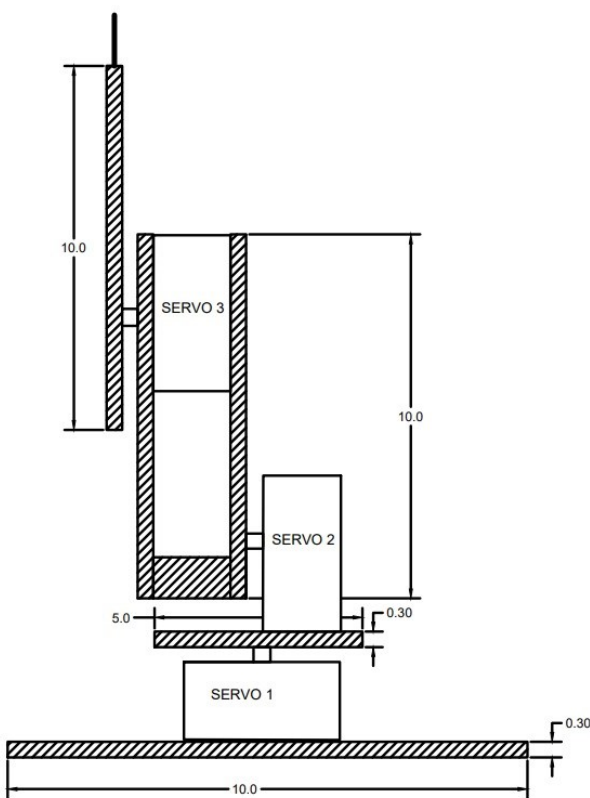


*Slika 4.2. Spoj komponenti bez konstrukcije.*

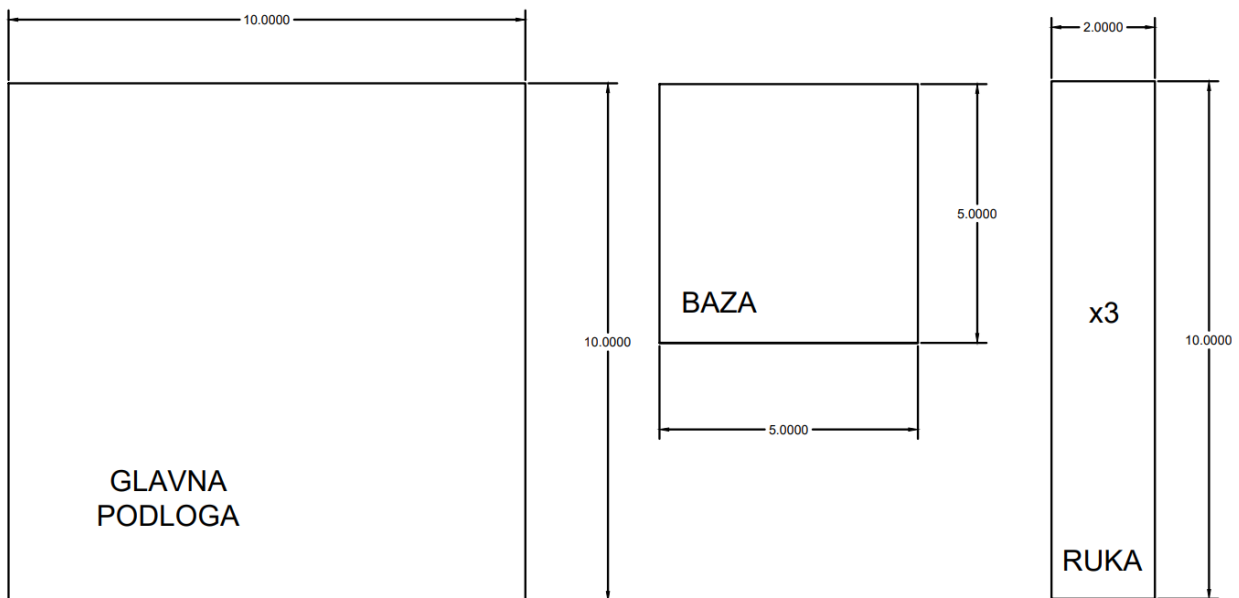


*Slika 4.3. Prikaz komponenti sa prednje strane.*

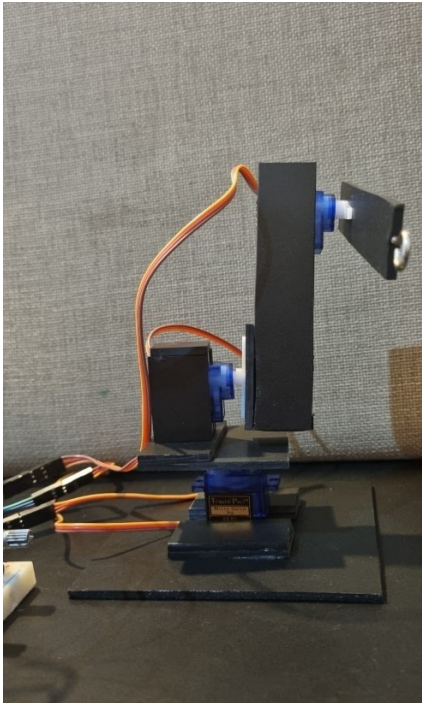
Nakon uspješnog spajanja svih elektroničkih komponenta potrebno je izraditi fizičku konstrukciju robotske ruke, prema nacrtu ,prikazanom na slici 4.4., i dimenzijama, prikazanima na slici 4.5. Svi dijelovi robotske ruke, koje smo oblikovali prema nacrtu, izrađeni su od izdržljivog crnog pleksiglasa. Nakon izrade dijelova, uslijedilo je precizno spajanje sa servo motorima. Prvi servo motor čvrsto je montiran na čvrstu podlogu, koja igra ključnu ulogu kao glavni oslonac ili baza robotske ruke. Ova baza ima zadatak pružiti stabilnu platformu kako bi se spriječilo prevrtanje robotske ruke tijekom njenog kretanja. Drugi servo motor, smješten iznad prvog, spojen je na podlogu koja se nalazi na prvom servo motoru. Na ovaj servo motor povezan je prvi segment robotske ruke, koji predstavlja prvi zglob. Drugi servo motor djeluje kao pokretna točka na kojoj se robotska ruka može savijati i kretati u različitim smjerovima. Na krajnjem dijelu prvog segmenta robotske ruke, nalazi se treći servo motor, prikazan na slici 4.6., koji obavlja ulogu drugog zgloba robotske ruke. Ovdje se postavlja završni segment ruke, koji služi kao mjesto, prikazano na slici 4.7., gdje se može pričvrstiti teret koji će robotska ruka podizati i pomicati.



Slika 4.4. Shematski prikaz servo motora.



*Slika 4.5. Prikaz dimenzija glavnih dijelova robotske ruke.*



*Slika 4.6. Prednji prikaz robotske ruke.*



*Slika 4.7. Bočni prikaz robotske ruke.*

## 5. PROGRAMSKI KOD ARDUINO UNO ZA ROBOTSKU RUKU

Nakon završetka izrade robotske ruke, potrebno je napisati odgovarajuću programsku podršku kako bi robotska ruka izvršavala željene funkcije.

Na samom početku programa potrebno je definirati biblioteku koja služi za upravljanje servo motorima. Nakon toga smo definirali servo motore te ulazne i izlazne pinove za LED diode, potencioetre i tipkala. Ovaj dio koda postavlja sve potrebne varijable i objekte za kontrolu servomotora, čitanje potencijometara, praćenje stanja tipkala i pohranu pozicija servomotora koje će se koristiti u glavnoj petlji programa. U glavnoj petlji, pozicije servomotora se ažuriraju na temelju trenutnih vrijednosti potencijometara, te ovisno o pritisnutim tipkalima izvršavaju se radnje koje su opisane u nastavku rada.

```
#include <Servo.h>

Servo servo1;
Servo servo2;
Servo servo3;

int LED1 = 2;
int LED2 = 3;
int LED3 = 4;
int LED4 = 5;
int LED5 = 6;

int button1 = 12;
int button2 = 13;

int button1Presses = 0;
boolean button2Pressed = false;

int pot1 = A0;
int pot2 = A1;
int pot3 = A2;

int pot1Val;
int pot2Val;
int pot3Val;
```

```
int pot1Angle;  
int pot2Angle;  
int pot3Angle;  
  
int servo1PosSaves[] = {0, 0, 0, 0, 0};  
int servo2PosSaves[] = {0, 0, 0, 0, 0};  
int servo3PosSaves[] = {0, 0, 0, 0, 0};
```

Funkcija *setup* se izvršava samo jednom prilikom uključanja ili resetiranja mikrokontrolera. Ovdje se postavljaju početna stanja i konfiguracije pinova i komponenti koje se koriste u ostatku programa. Postavljeni su izlazni pinovi za servo motore preko kojih će Arduino upravljati s njihovim pozicijama. Također, pinovi na koje su spojene LED diode postavljeni su kao izlazni pinovi, dok pinovi na koje su spojena tipkala su postavljeni kao ulazni pinovi. Na kraju je postavljena komunikacija s računalom na brzinu od 9600 bitova po sekundi.

```
void setup() {  
    servo1.attach(9);  
    servo2.attach(10);  
    servo3.attach(11);  
  
    pinMode(LED1, OUTPUT);  
    pinMode(LED2, OUTPUT);  
    pinMode(LED3, OUTPUT);  
    pinMode(LED4, OUTPUT);  
    pinMode(LED5, OUTPUT);  
    pinMode(button1, INPUT);  
    pinMode(button2, INPUT);  
  
    Serial.begin(9600);  
}
```

Funkcija *loop* u Arduino programu predstavlja srce programa i neprestano se izvršava. U ovom dijelu koda čitaju se vrijednosti s tri potencijometra (*pot1*, *pot2* i *pot3*). Svaki potencijometar omogućuje korisniku ručno podešavanje položaja odgovarajućem servo motora, pomicanjem potencijometra rezultira promjenom napona na odgovarajućem analognom pinu Arduino ploče. Nakon što se očitaju te vrijednosti, koristi se funkcija *map* kako bi preslikala te očitane vrijednosti iz raspona 0-1023 u raspon 0-179. Svaka očitana vrijednost s potencijometra se mapira u specifični kut, koji će servo motor zauzeti. Zatim, koristeći te izračunate kutove, funkcija postavlja poziciju svakog servo motora na odgovarajući kut. Ovo omogućuje servo motorima da se dinamički prilagode trenutnom položaju potencijometara, što znači da će se svaki servo motor ponašati prema trenutnom položaju pripadajućeg potencijometra.

```
void loop() {  
    pot1Val = analogRead(pot1);  
    pot1Angle = map(pot1Val, 0, 1023, 0, 179);  
    pot2Val = analogRead(pot2);  
    pot2Angle = map(pot2Val, 0, 1023, 0, 179);  
    pot3Val = analogRead(pot3);  
    pot3Angle = map(pot3Val, 0, 1023, 0, 179);  
    servo1.write(pot1Angle);  
    servo2.write(pot2Angle);  
    servo3.write(pot3Angle);  
}
```

U sljedećem dijelu koda prati se stanje tipkala (*button1*) i izvršavaju odgovarajuće radnje kada je tipkalo pritisnuto. Program provjerava je li tipkalo pritisnuto, te dodaje kratku pauzu od 100 ms za softverski *debouncing*. Varijabla *button1Presses* se povećava za 1 svaki put kad je tipkalo pritisnuto. Naredba *switch* se koristi za provjeru vrijednosti varijable *button1Presses* i izvođenje odgovarajućih radnji. Prilikom prvog pritiska tipkala pohranjuju se kutovi servomotora, LED dioda se uključuje za označavanje pohranjene prve pozicije, te se ispisuje poruka na serijskom monitoru za potvrdu pohranjivanja prve pozicije. Na ovom principu omogućena je pohrana pet pozicija servomotora pritiscima na prvo tipkalo, uz vizualnu identifikaciju pohranjenih pozicija putem pripadne LED diode.

```

if (digitalRead(button1) == HIGH) {
    delay(100);
    button1Presses++;

    switch (button1Presses) {
        case 1:
            servo1PosSaves[0] = pot1Angle;
            servo2PosSaves[0] = pot2Angle;
            servo3PosSaves[0] = pot3Angle;
            digitalWrite(LED1, HIGH);
            Serial.println("Pos 1 Saved");
            break;
        case 2:
            servo1PosSaves[1] = pot1Angle;
            servo2PosSaves[1] = pot2Angle;
            servo3PosSaves[1] = pot3Angle;
            digitalWrite(LED2, HIGH);
            Serial.println("Pos 2 Saved");
            break;
        case 3:
            servo1PosSaves[2] = pot1Angle;
            servo2PosSaves[2] = pot2Angle;
            servo3PosSaves[2] = pot3Angle;
            digitalWrite(LED3, HIGH);
            Serial.println("Pos 3 Saved");
            break;
        case 4:
            servo1PosSaves[3] = pot1Angle;
            servo2PosSaves[3] = pot2Angle;
            servo3PosSaves[3] = pot3Angle;
            digitalWrite(LED4, HIGH);
            Serial.println("Pos 4 Saved");
            break;
        case 5:
            servo1PosSaves[4] = pot1Angle;
            servo2PosSaves[4] = pot2Angle;
            servo3PosSaves[4] = pot3Angle;
            digitalWrite(LED5, HIGH);
            Serial.println("Pos 5 Saved");
            break;
    }
}

```



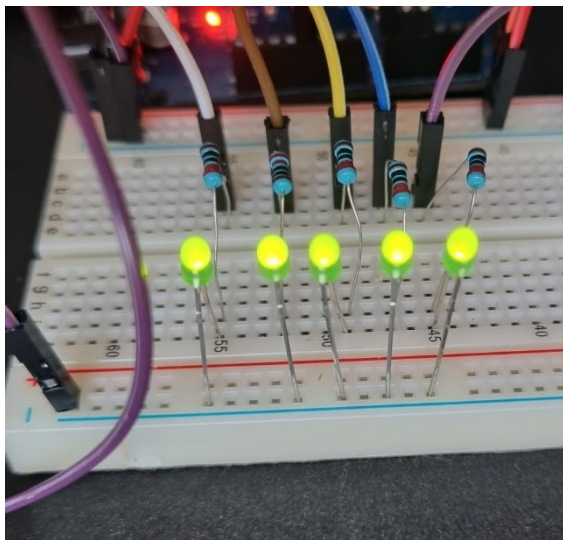
U sljedećem dijelu koda provjerava ako je tipkalo dva (*button2Pressed*) pritisnuto. Ako je pritisnuto prvo se postavlja varijabla *button2Pressed* na vrijednost logike istine kao oznaku da je tipkalo pritisnuto. Nakon toga, unutar *if* naredbe nalazi se *for* petlja koja se izvršava pet puta gdje se svaki put postavljaju nove pozicije servo motora koje su pohranjene vrijednosti iz nizova sa početka koda *servo1PosSaves*, *servo2PosSaves* i *servo3PosSaves*. Tijekom petlje, trenutni kutovi potencijometara ispisuju se na serijskom monitoru. Nakon svake iteracije petlje, program čeka 2 sekunde prije nego što nastavi s idućom iteracijom. Nakon završetka petlje, program dodatno čeka 300 ms prije ponovnog izvođenja, te se petlja izvodi dok se Arduino pločica ne resetira.

```
if (digitalRead(button2) == HIGH) {
    button2Pressed = true;
}
if (button2Pressed) {
    for (int i = 0; i < 5; i++) {
        servo1.write(servo1PosSaves[i]);
        servo2.write(servo2PosSaves[i]);
        servo3.write(servo3PosSaves[i]);
        Serial.println(" potentiometer Angles: ");
        Serial.println(servo1PosSaves[i]);
        Serial.println(servo2PosSaves[i]);
        Serial.println(servo3PosSaves[i]);
        delay(2000);
    }
}
delay(300);
}
```

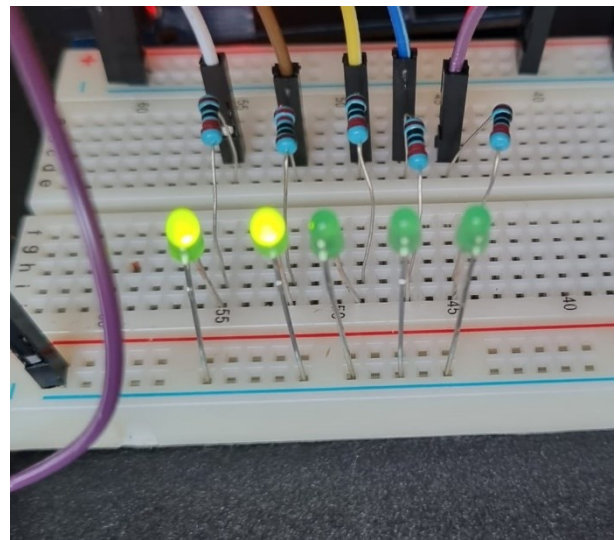
## 6. PRINCIP RADA ROBOTSKE RUKE

Početak rada ove robotske ruke započinje priključivanjem Arduino pločice na napajanje putem USB priključka na laptopu. Nakon što je pločica napajana, servo motori koji kontroliraju ruku postavljaju se u svoje početne pozicije kao inicijalna konfiguracija. Ova robotska ruka zamišljena je kao model robotske ruke u industriji i omogućuje korisniku spremanje i izvođenje do pet različitih pozicija ruke.

Pomicanjem potenciometara, korisnik istovremeno upravlja odgovarajućim servo motorima, što omogućuje precizno pozicioniranje ruke. Kada želimo pohraniti trenutnu poziciju, potrebno je pritisnuti prvo tipkalo, što rezultira spremanjem pozicije i aktiviranjem prve LED diode za označavanje pohranjene pozicije, prikazano na slici 6.2. Ovaj postupak može se ponoviti do pet puta, a svaka spremljena pozicija označena je odgovarajućom LED diodom. Kada svih pet LED dioda svijetli, prikazano na slici 6.1., to ukazuje da su sve moguće pozicije pohranjene.



*Slika 6.1. Stanje dioda prilikom pohrane svih pozicija.*



*Slika 6.2. Stanje dioda prilikom pohrane dvije pozicije.*

Nakon što su pozicije pohranjene, korisnik može pokrenuti izvođenje tih pozicija pritiskom na drugu tipku. Robotska ruka će zatim izvršavati pohranjene pozicije kontinuirano. Ako želi zaustaviti izvođenje ili resetirati ciklus, korisnik može pritisnuti tipku "Restart" na Arduino pločici, što će ponovno pokrenuti program.

## 7. ZAKLJUČAK

Tijekom izrade završnog rada sam se intenzivno upoznao s Arduino platformom i njenim razvojnim okruženjem (Arduino IDE). Kroz programski kod koji sam napisao za robotsku ruku, stekao sam dublje razumijevanje rada Arduino pločice. Važno je napomenuti da Arduino omogućava izradu različitih sustava, bilo jednostavnijih ili složenijih, što doprinosi praktičnoj primjeni u svakodnevnom životu.

Automatizacija je prisutna u mnogim aspektima svakodnevnog života te donosi mnoge prednosti. Arduino se ističe kao iznimno koristan i efikasan alat za ubrzanje razvoja automatiziranih sustava. Iako je ovaj projekt prikazao samo osnovne mogućnosti Arduina u automatizaciji, otvara vrata mnogim drugim primjenama.

Tijekom izrade ovog projekta sam stekao nova znanja o spajanju različitih komponenata i razumijevanju njihovog rada. Korištenjem potencijometara i mikro servomotora omogućio sam precizno upravljanje pozicijama servomotora na temelju vrijednosti potencijometara.

Važno je napomenuti da izrada robotske ruke nije bila izrazito kompleksna niti skupa jer su korišteni materijali i komponente široko dostupni. Unatoč manjim izazovima, poput problema s kontaktima, rješavanjem tih problema uspješno sam dovršio robotsku ruku.

Osnovna ideja ovog projekta bila je stvoriti funkcionalnu robotsku ruku s tri stupnja slobode koja može pohraniti do pet različitih pozicija i neprestano ih izvršavati. Ideja je uspješno realizirana i cilj je postignut.

## Literatura

- [1] „What is an arduino“, s interneta, <https://learn.sparkfun.com/tutorials/what-is-an-arduino>,
- [2] „Servomotor“, s interneta, <https://en.wikipedia.org/wiki/Servomotor>,
- [3] „Potenciometar“, s interneta, <https://hr.wikipedia.org/wiki/Potenciometar>,
- [4] „Rotary potentiometer“, s interneta, <https://core-electronics.com.au/rotary-potentiometer-250k-ohm-logarithmic-panel-mount.html>,
- [5] „Linearni potenciometar“, s interneta, <https://soldered.com/hr/proizvod/klizni-potenciometar/>,
- [6] „Robotic arm in automotive industry“, s interneta, <https://www.finance-monthly.com/2021/10/a-look-at-how-robots-are-used-in-the-automotive-industry/>,
- [7] „Robotic arm“, s interneta, <https://www.intel.com/content/www/us/en/robotics/robotic-arm.html>,
- [8] „Robotic arm in healthcare industry“, s interneta <https://www.automate.org/news/3-powerful-ways-to-deploy-robotic-arms-in-healthcare-industry>,
- [9] „PN spoj kod diode“, s interneta <https://puaks.blogspot.com/2014/02/dioda.html>,
- [10] „Robotska ruka za zavarivanje“, s interneta <https://www.isystemsweb.com/applications/robotic-weld-cell-filtration/>,
- [11] „Pushbutton“, s interneta <https://www.teachwithict.com/pushbttton.html>,
- [12] „Konstrukcija robotske ruke“, s interneta <https://www.universal-robots.com/products/ur16-robot/>,
- [13] „Podjela robotskih ruku“, s interneta <https://www.universal-robots.com/in/blog/robotic-arm/>,
- [14] „Cartesian robotic arm“, s interenta <https://www.linearmotiontips.com/what-is-a-cartesian-robot/>,
- [15] „Cylindrical Robotic arm“, s interneta <https://www.hitbotrobot.com/product/z-arm-2442e/>
- [16] „SCARA robotic arm“, s interneta <https://epson.com/scara-robots-product-family>,

- [17] „Tower Pro servo motor“, s interneta <https://soldered.com/hr/proizvod/servo-motor-towerpro-sg90/>,
- [18] „Duty cycle“,s interneta <https://learn.sparkfun.com/tutorials/pulse-width-modulation/duty-cycle>,
- [19] „PWM signal“, s interneta <https://electronicscoach.com/pulse-width-modulation.html>,
- [20] „Tower Pro servo PWM signal“, s interneta <https://www.engineersgarage.com/servo-motor-sg90-9g-with-89c51-microcontroller/>,
- [21] „Dioda“, s interneta <https://soldered.com/hr/proizvod/led-dioda-10mm-zuta/>

## Sažetak i ključne riječi

Robotska ruka s tri stupnja slobode predstavlja ključni element u širem kontekstu automatizacije u industriji, što je glavna tema ovog završnog rada. Robotske ruke imaju značajnu ulogu u suvremenim industrijskim sustavima, posebno u proizvodnim okruženjima. Ovaj rad se usmjerava na konkretan primjer robotske ruke koja se kontrolira putem potenciometara i koristi servo motore za simulaciju rada u proizvodnom procesu. Kroz pomicanje potenciometara, mijenjaju se vrijednosti koje se direktno prenose na servo motore, što rezultira njihovim odgovarajućim pokretima. Ključni dio projekta je mogućnost pohranjivanja pozicija servomotora pritiskom na prvu tipku. Nakon što se pohrani do pet različitih pozicija, drugom tipkom pokreće se simulacija rada robotske ruke. Također postoji fleksibilnost u smislu pohranjivanja manjeg broja pozicija, što omogućava da se preostale pozicije tretiraju kao stanje mirovanja. Osim toga, tipka za ponovno pokretanje na Arduino pločici služi za resetiranje cijelog sustava i povratak na početno stanje.

Ključne riječi: Robotska ruka, Arduino pločica, servo motori, potenciometri, tipkala, automatizacija,

## **Summary and key words**

The robotic arm with three degrees of freedom represents a key element in the wider context of the industrial automation, which is the main theme of this bachelor's thesis. Robotic arms play a significant role in the modern industrial systems, particularly in the manufacturing environments. This paper focuses on a specific example of a robotic arm controlled by the potentiometers with three servo motors to simulate operations within a manufacturing process. By manipulating the potentiometers, adjusted values are transmitted to the servo motors, resulting in the corresponding movements. The main feature of this specific robotic arm is the ability to store positions of the servo motors by pressing the first button. Once up to five different positions are stored, the second button is used to initiate a simulation of the robotic arm's operation. Furthermore, there is flexibility in terms of storing a smaller number of positions, allowing the remaining positions to be treated as a resting states. Additionally, the restart button on the Arduino board serves to reset the entire system and return it to its initial state.

Keywords: Robotic Arm, Arduino Board, Servo Motors, Potentiometers, Buttons, Automation.