

Upravljanje strojevima u proizvodnji korištenjem Arduino platforme: projekt robotske ruke

Kalajžić, Dominik

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:231:000539>

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

Download date / Datum preuzimanja: **2025-03-26**

Repository / Repozitorij:

[Repository of the University of Rijeka University Studies, Centers and Services - RICENT Repository](#)



SVEUČILIŠTE U RIJECI

Studij politehnike

Sveučilišni preddiplomski studij politehnike

Dominik Kalajžić

**Upravljanje strojevima u proizvodnji
korištenjem *Arduino* platforme: projekt
robotske ruke**

Završni rad

Mentor: izv. prof. dr. sc. Damir Purković

Rijeka, 2023.

SVEUČILIŠTE U RIJECI

Studij politehnike

Studijski program: sveučilišni preddiplomski studij politehnike

Dominik Kalajžić

Mat. broj: 9998000976

**Upravljanje strojevima u proizvodnji pomoću Arduino razvojne
platforme: projekt robotske ruke**

-završni rad-

Mentor: izv. prof. sc. Damir Purković

Rijeka, 2023.

Zadatak za završni rad

Pristupnik: **Dominik Kalajžić**

Naziv završnog rada: **Primjena Arduino razvojne platforme za upravljanje industrijskim strojevima**


Naziv završnog rada na eng. jeziku: **Application of Arduino development platform for industrial machinery management**

Sadržaj zadatka:

Rješenjem zadatka je potrebno obuhvatiti sljedeće:

1. Uvodni dio – o industrijskim strojevima, tehnologiji upravljanja i problemima edukacije;
2. Razvojne platforme – svrha, vrste i posebnosti, primjena i mogućnosti;
2. Arduino razvojna platforma – sklopovlje, razvojno sučelje, mogućnosti i ograničenja;
3. Razrada projekta primjene – cilj, razrada ideje, izrada i sastavljanje, predstavljanje;
4. Mogućnosti, ograničenja i značaj primjene projekta za obrazovanje inženjera;
5. Zaključak.

Mentor: doc. dr. sc. Damir Purković




(potpis mentora)

Voditelj za završne radove



Zadatak preuzet: 27. ožujka 2023.



(potpis pristupnika)

IZJAVA

Izjavljujem da sam završni rad izradio samostalno, isključivo znanjem stečenim na Studiju politehnike Sveučilišta u Rijeci, služeći se navedenim izvorima podatka i u stručno vodstvo mentora izv. prof. dr. sc. Damira Purkovića.

Zahvaljujem se svom mentoru izv. prof. dr. sc. Damiru Purkoviću koji mi je omogućio sve potrebne izvore i stručne savjete kojima mi je pomogao u izradi ovog završnog rada i što je imao strpljena i vremena za sva moja pitanja.

Zahvaljujem svim svojim prijateljima koji su uvijek bili uz mene i pomogli mi tijekom studiranja.

Zahvaljujem i svojoj obitelji koja me podržala, najviše svojim roditeljima koji su uvijek bili tu za mene, bez obzira na sve.

U Rijeci, 30.07.2023. godine

Dominik Kalajžić

Sažetak

Danas čovječanstvo živi na pragu nove industrijske revolucije koja će, kao i prijašnje, uvelike utjecati na njegov život i rad. Automatizacija proizvodnje sa sobom nosi nove izazove za sadašnje i buduće inženjere pogotovo na STEM području. U ovom završnom radu obrađene su vrste i klasifikacije industrijski strojeva u raznim granama proizvodnje kao i načine upravlja tim industrijskim strojevima. Zatim su obrađeni izazovi i postupci obrazovanja inženjera uz pomoć razvojnih platformi, primarno Arduino UNO platforme. Postoje mnoge vrste razvojnih platformi od kojih su najpoznatiji *Arduino* i *Raspberry Pi*, stoga ćemo te razvojne platforme obraditi u ovom završnom radu. Potom je razrađen projektni zadatak Robotske ruke. Robotska ruka sastavljena je od aluminijskih nosača i pokreće je servo motora. Robotom se upravlja putem Android aplikacije koristeći Bluetooth modul. Možemo programirati unaprijed isplanirane pokrete, kao npr. Uzimanje raspoređenog tereta koji potom robot slaže jedan na drugoga ili upravljati individualnim dijelovima robota. Projektni zadatak uključuje detaljnu razradu dijelova koje su korišteni te opširnije o samim robotima u proizvodnji i dijelovima koji se koriste za izradu istih. Potom je objašnjeno kako spojiti robota kako bi mogli njime upravljati putem Arduino UNO platforme, odnosno mobilne aplikacije putem Bluetootha. Na kraju su navedene moguće domene korištenja robotske ruke i vlastiti osvrt o budućnosti pojedinih elemenata u industriji.

Ključne riječi: automatizacija, industrijski strojevi, Arduino, robotska ruka.

Managing machinery in production using Arduino development platform: robot arm project

Abstract

Today, humanity lives on the threshold of a new industrial revolution, which, like the previous ones, will greatly affect its life and work. Production automation brings with it new challenges for current and future engineers, especially in the STEM field. In this final paper, we will deal with the types and classifications of industrial machines in various branches of production, as well as the ways of managing these industrial machines. Then we address the challenges and ways of educating engineers with the help of development platforms, primarily Arduino UNO. There are many types of development platforms, the most famous of which are Arduino and Raspberry Pi, so we will cover these development platforms in this final paper. Then the project task of the Robot Hand was elaborated. The robotic arm is made of aluminium supports and is driven by servo motors. The robot is controlled via Android application using a Bluetooth module. We can program pre-planned movements, such as picking up distributed loads that the robot stacks on top of each other, or after that, the development of the project work, robotic hands, begins. The project assignment includes a detailed elaboration of the parts that we will use and in more detail about the robots themselves in production and the parts that are used to make them. Then it was explained how to connect the robots so that they could be controlled via the Arduino Uno platform, that is, the mobile application via Bluetooth. At the end, the possible domains of use of the robotic arm and our own review of the future of certain elements in the industry are listed.

Key words: automation, industry machines, Arduino, robotic arm.

SADRŽAJ:

| | |
|--|-----------|
| 1. Uvod | 8 |
| 1.1. Klasifikacija industrijskih strojeva | 8 |
| 1.1.1. Prijenosni strojevi | 9 |
| 1.1.2. Laki strojevi | 10 |
| 1.1.3. Srednje teški strojevi | 10 |
| 1.1.4. Teški strojevi | 11 |
| 1.2. IoT (Internet of Things) | 11 |
| 1.3. Prednosti i nedostaci IoT-a | 13 |
| 1.4. PLC sustavi | 14 |
| 2. Razvojne platforme | 16 |
| 2.1. Uloga razvojnih platformi u obrazovanju | 15 |
| 2.2. Klasifikacija razvojnih platformi | 15 |
| 2.2.1. Raspberry Pi | 17 |
| 2.2.2. Prednosti razvojne platforme Raspberry Pi | 18 |
| 2.2.3. Nedostaci razvojne platforme Raspberry Pi | 18 |
| 2.3. Arduino | 18 |
| 2.3.1. Prednosti Arduino razvojne platforme | 19 |
| 2.3.2. Nedostaci Arduino razvojne platforme | 20 |
| 3. Robotska ruka | 21 |
| 3.1. Kartezijanski robot (linearni robot) | 22 |
| 3.2. Kolaborativni robot | 22 |
| 3.3. Cilindrični robot | 24 |
| 3.4. Sferični robot | 25 |
| 3.5. SCARA roboti | 26 |
| 3.6. Zglobni robot | 27 |
| 3.7. Paralelni robot | 27 |
| 3.8. Servo motori | 28 |
| 4. Projekt – Razvoj robotske ruke | 29 |
| 4.1. Komponente robotske ruke | 29 |

| | |
|--|-----------|
| 4.2. HC-06 Bluetooth modul | 30 |
| 4.3. Memorijski modul (AT24C256 EEPROM) | 31 |
| 4.4. Proširenje Adafruit PCA9685 | 32 |
| 4.5. Arduino UNO | 33 |
| 4.6. Programiranje komponenti robotske ruke | 33 |
| 4.6.1. Deklaracija LCD zaslona | 36 |
| 4.6.2. Deklaracija EEPROM memorije | 39 |
| 4.6.3. Bluetooth deklaracija | 40 |
| 4.6.4. Deklaracija servo kontrolera | 40 |
| 4.6.5. Aplikacija za upravljanje robotskom rukom | 42 |
| 4.7. Postupak spajanja | 43 |
| 5. Mogućnosti primjene projekta | 44 |
| 6. Zaključak | 45 |
| 7. Literatura | 46 |
| 8. Popis slika | 48 |

1. Uvod

Automatizacija proizvodnje je jedna od najvećih prekretnica u povijesti proizvodnje. Prelazak sa klasičnih načina povezivanja na umrežavanje pojedinačnih sustava i sustava u cjelini pomoću 5G mreža predstavlja nove izazove za inženjere komunikacije i automatizacije. Industrijski strojevi su upravljani i sinkronizirani korištenjem sustava PLC (*Programmable Logic Controller*), IoT (*Internet of Things*) i sustavima za upravljanje proizvodnjom. PLC je industrijsko računalo koje je ojačano i prilagođeno za kontrolu proizvodnih procesa ili bilo kojeg procesa koji zahtjeva visoku pouzdanost, jednostavnost programiranja i dijagnoza grešaka u procesu [14]. Temelji se na mikroprocesoru s programibilnom memorijom koja se koristi za pohranu programskih uputa i raznih funkcija [14]. IoT uređaj može biti i obična žarulja u kući ako se može uključiti pomoću aplikacije za pametni telefon ili termostata kojim se pomoću neke aplikacije upravlja svakim danom i podešava željena temperatura stana[1]. Ovakvi sustavi mogu se primijeniti na različitim stvarima, primjerice od dječjih igračaka do autonomnih automobila. Neki veći objekti mogu biti prepuni malih IoT pametnih sustava, kao što je mlazni motor aviona koji se sastoji od tisuća senzora za prikupljanje i slanje podataka kako bi se rad motora mogao analizirati [1]. Svrha ovog završnog rada je opisati industrijske strojeve koji se danas koriste u proizvodnji i kojima se upravlja putem sustava automatizacije, kao i način i sustav upravljanja strojevima. Zatim govorimo o korištenju razvojnih platformi i njihovim prednostima i nedostacima u obrazovanju inženjera. Razvojne platforme su idealan temelj za uvod u industriju strojeva i njihovog upravljanja jer obuhvaćaju sva područja znanosti s kojima se budući inženjeri susreću. Za projektni zadatak uzeli smo robotsku ruku jer odlično služi kao uvod u industrijske strojeve današnjice i zbog same kompleksnosti projekta, od sastavljanja, programiranja i upravljanja robotskom rukom.

1.1. Klasifikacija industrijskih strojeva

Industrijski strojevi su bilo koji strojevi ili oprema koje koristi proizvođač u proizvodnoj ustanovi. Postoji više načina klasificiranja industrijskih strojeva:

1. Po veličini:
 - a. prijenosni strojevi,

- b. laki strojevi,
- c. srednje teški strojevi,
- d. teški strojevi.

2. Po metodi djelovanja:

- a. ručno upravljani strojevi,
- b. polu-automatski strojevi,
- c. automatski strojevi.

3. Po tipu proizvodu ili industrije:

- a. tekstilna industrija,
- b. strojevi u agrokulturi,
- c. proizvodnja naftnih derivata i prerada,
- d. metalna industrija,
- e. industrija papira,
- f. industrija obrade hrane,
- g. energetska industrija,
- h. farmaceutska industrija,
- i. industrija plastike i PVC-a,
- j. industrija građevinskih materijala.

4. Po načinu pogona:

- a. strojevi pogonjeni električnim motorima,
- b. strojevi pogonjeni motorima,
- c. hidraulični strojevi,
- d. pneumatski strojevi,
- e. pogonjeno energijom vjetra ili vode,
- f. pogonjeno solarnom energijom,

Potrebno je ispravno klasificirati strojeve kako bi lakše razumjeli njihovu namjenu u industriji proizvodnje, i njihovu izvedbu. Potom će se u nastavku rada elaborirati karakteristike po kojima se dijele, odnosno njihovoj težini i funkciji.

1.1.1. Prijenosni strojevi

U ovu kategoriju prijenosnih strojeva (slika 1.1.) se ubrajaju svi strojevi koje možemo prenositi sami bez korištenja transportne opreme i ne teže više od 20 kilograma, kao što su mjerni instrumenti (kaliperi i ostali mjerni instrumenti, brusilice, bušilice i slični alat za ručnu obradu materijala.



Slika 1.1. Prijenosni strojevi

1.1.2. Laki strojevi

U ovu kategoriju spadaju strojevi (slika 1.2.) koji teže manje od jedne metričke tone. Primjer su strojevi za prešanje, tokarski strojevi, strojevi za glodanje, strojevi za bušenje, robotske ruke i slično.



Slika 1.2. Laki stroj

1.1.3. Srednje teški strojevi

U ovu kategoriju spadaju industrijski strojevi (slika 1.3.) koji teže između jedne i 10 metričkih tona. Primjer su: CNC tokarski strojevi, Indukcijske peći, industrijska robotska oprema, bojleri, strojevi za injekcijsko prešanje, VMC stroj i slično.



Slika 1.3. Srednje teški strojevi

1.1.4. Teški strojevi

U ovu kategoriju spadaju strojevi koji teže više od 10 tona. Primjer su strojevi koji se koriste za ekstrakciju nafte i zemnog plina i strojevi za rudarenje.

1.2. IoT (Internet of Things)

IoT(eng. *Internet of Things*) naziv je za milijarde fizičkih uređaja u čitavom svijetu koji su povezani s internetom, a svi prikupljaju i dijele podatke. Zbog jeftine proizvodnje mikrokontrolera i pristupačnog bežičnog interneta bilo koji uređaj ili stroj može postati IoT. Povezivanjem različitih objekata i dodavanjem senzora dolazi do „digitalne inteligencije“ (komunikacija između uređaja bez posredovanja čovjeka) uređaja, omogućujući im komuniciranje i razmjenu podataka u stvarnom vremenu bez interakcije čovjeka, iako ljudi mogu komunicirati i upravljati uređajem[1]. IoT se naziva sljedećom industrijskom revolucijom, promijeniti će način interakcije sa fizičkim svijetom za sve poslove, države i potrošače. Do 2020., godine, u svijetu će biti više od 34 bilijun uređaja povezanih sa internetom. [2] To je mreža svih mreža koja se sastoji od milijuna privatnih, javnih, akademskih poslovnih i državnih mreža, od lokanog do globalnog opsega, koje su povezane širok raspon elektroničkih, bežičnih i optičkih mrežnih tehnologija.

Za uspješnu implementaciju Interneta Stvari postoje određeni preduvjeti[2]:

1. Dinamička potražnja resursa – mora biti sposoban efikasno upravljati i raspodijeliti električnu energiju kako bi produžio trajanje baterije, odnosno uštedio energiju;
2. Potrebe u stvarnom vremenu – praćenje i prikupljanje podataka u stvarnom vremenu koja se šalju centralnom računalu;
3. Eksponencijalni rast potražnje – IoT sustav se mora moći nadograđivati zbog konstantnog porasta broja uređaja i podataka;
4. Dostupnost aplikacija – mora biti široko dostupna i pouzdana aplikacija;
5. Zaštita podataka i privatnosti korisnika – potrebna je visoka razina zaštite jer čuva osjetljive podatke korisnika;
6. Učinkovita potrošnja energije aplikacija – potrebna je visoka energetska učinkovitost;
7. Izvršenje aplikacija u blizini krajnjih korisnika – izvršavanje na samim uređajima kako bi se smanjila latencija i vrijeme reakcije;
8. Pristup otvorenom i interoperabilnom sustavu oblaka – mogućnost povezivanja centralnog računala sa ostalim platformama i *cloud* memorijama.

Prema drugom autoru, postoje tri komponente potrebne za uspješan rad Interneta stvari[2]:

- Hardver – sastavljen od senzora, aktuatora, IP kamera, CCTV-a i ugrađenih komunikacijskog hardvera
- Middleware – pohrana na zahtjev i računalni alati za analitiku podataka s oblakom i velikim podacima
- Analitika – jednostavni alati za vizualizaciju i interpretaciju koji se mogu dizajnirati za različite primjene

1.3. Prednosti i nedostaci IoT-a

Nakon obrade IoT-a u nastavku će se govoriti o prednostima i nedostacima implementacije IoT sustava u industrijskoj proizvodnji i raznim aparatima koji okružuju čovjeka.

Prednosti IoT-a su [1]:

1. Ušteda troškova: IoT čini naše sustave efikasnijim. Omogućuje da elektronički uređaji međusobno učinkovito komuniciraju čime štede energiju.
2. Omogućuje razmjenu i prikupljanje podataka za daljnju analizu i obradu te se tako smanjuju troškovi proizvodnje u industrijama.
3. Informacije: IoT može prikupljati bitne informacije o sustavu unutar industrije te je moguće odraditi preventivne servise koji će nas kasnije finansijski koštati jeftinije od zastoja cijelog postrojenja.
4. Automatizacija i upravljanje: Svi strojevi mogu međusobno komunicirati čime se postiže poboljšavanje u sustavima automatizacije i samog upravljanja. S obzirom na prisutnost bežične komunikacije koja je svakim danom kvalitetnija, sustavi automatizacije također napreduje.
5. Povećanje produktivnosti: Produktivnost ima ključnu ulogu u profitabilnosti bilo kojeg posla. IoT poboljšava efikasnost rada i također smanjuje neusklađenost susta te time smanjuje i same troškove.

Nedostaci IoT-a su [1]:

1. Život ovisan o tehnologiji: Potpuni oslonac na IoT sustave i uređaje može stvoriti probleme u slučaju zatajenja ili pada infrastrukture.
2. Privatnost i sigurnost: Hakerski napadi današnja su svakodnevnicu u IoT svijetu što otežava čuvanje podataka od neovlaštenih korisnika.
3. Manje radnih mjesta: U svakodnevnicu sve se više sustava automatizira gdje IoT također ima veliki doprinos. Takvim pristupom prirodno je da se zahtjevi za ljudskim resursima smanjuju, a kvalificirane radne snage sve je manje.
4. Složenost: Pojava kvarova, bili ono softverskog ili hardverskog tipa, na IoT sustavima kojih je sve više, mogla bi biti sve češća.
5. Kompatibilnost: Međunarodni standard za kompatibilnost IoT-a trenutno ne postoji. Implementacija različitih rješenja proizvođača mogu otežati međusobnu komunikaciju uređaja.

IoT sustavi se sve više pojavljuju u svakodnevnim životima ljudi. Takvi sustavi upravljanje uređajima i praćenje stanja aparata činim pogodnijim. Sa tom razinom umreženosti javlja se problem krađe podataka i zadiranja u privatnost ljudi. Umreženost i automatizacija u industriji unapređuje profitabilnost i metode održava, ali u isto vrijeme smanjuje broj potrebne radne snage.

1.4. PLC sustavi

PLC (*Programmable Logic Controller*) (slika 1.4.) je digitalni kompjuter koji se koristi za automatizaciju za tipično industrijske elektromehaničke procese, kao što su naprimjer upravljanje strojevima na proizvodnim trakama. PLC se koriste u mnogim strojevima, mnogih industrija. Dizajnirani su za višestruki aranžman digitalnih i analognih ulaza i izlaza, proširenih temperaturnih raspona, otpornosti na električne smetnje i otpornost na vibracije i udarce [2].

Prednosti PLC sustava su [9]:

- Fleksibilnost,
- Brži odziv,
- Jednostavnije spajanje i manje žica,

- Nema pokretnih dijelova,
- Modularni dizajn – laki popravci i pristup,
- Lako rukovanje sofisticiranijim i kompleksnijim sustavima,
- Niska cijena,
- Mogućnost dijagnosticiranja kvara.

Nedostaci PLC sustava[10]:

- Zahtjevno spajanje žica,
- PLC tehnologija zahtjeva edukaciju,
- Osjetljivost na visoke temperature i vibracije,
- Proizvođači PLC sustava proizvode samo sustave sa zatvorenom petljom,
- Prilikom korištenja PLC-a, u slučaju greške, zastoje traje sve dok se ne ukloni smetnja,
- Potrebno je dodati više operacijskih modula kako bi se maksimizirala fleksibilnost i performansa.



Slika 1.4. PLC moduli

Premda jednostavni za spajanje i rukovanje, najveći problem kod PLC modula je osjetljivost na visoke temperature i vibracije koje su u industriji u kojoj se primjenjuju veoma česte. Za rukovanje PLC modulima potrebno je iskustvo i edukacija koja ubrzava sposobnost otklona kvarova jer u suprotnom sustav proizvodnje je u zastoju.

2. Razvojne platforme

U nastavku će se opisati razvojne platforme i njihove uloge u pripremi i obrazovanju inženjera. Govoriti će se o njihovim prednostima i nedostacima te mogućim idejama o unapređenju sustava razvojnih platformi. Postoje mnoge razvojne platforme: ESP32 (*Espressif Systems*), Teensy 4.0 (sparkfun), Particle Boron, Beagle board i slične. Daleko najpopularnije su Arduino i RaspberryPi. Razlozi za takvu popularnost su: veoma niske cijene u odnosu na ostale razvojne platforme, u početku su izrađivani za potrebe edukacije studenata STEM područja, modularni i može ih razumjeti i koristiti apsolutni početnici.

2.1. Uloga razvojnih platformi u obrazovanju

Inženjerstvo se tradicionalno poučavalo deduktivno, s nastavnikom koji je uvodio temu izlažući relevantnu teoriju i matematičke modele, nakon čega su slijedile demonstracije podijeljene na dijelove koje primjenjuju teoretske koncepte za rješavanje nekih problema. Ciklus se obično završava procjenom sposobnosti studenata da odgovore na slična pitanja na svojim ispitima[12]. Izmijenjeni gospodarski, tržišni i politički uvjeti nameću obrazovnim institucijama zahtjeve i potrebe za razvojem organizacije koja će biti u mogućnosti efikasno se nositi sa vrlo zahtjevnim, promjenjivim i kontinuiranim razvojem industrije i tehnologije[13]. Stoga dolazi do potrebe za izmjenom kurikuluma i pristupa podučavanju budućih inženjera. U nastavku ćemo govoriti o jednom od pristupa, praktični radovi i razvojne platforme.

Arduino je platforma za učenje programiranja i korištenje mikrokontrolera. Osmišljena je kao (*Open Source*), odnosno platforma tzv. „Otvorenog koda“ što znači da ima slobodnu razmjenu hardverskih i softverskih komponenti te programskog koda. Uz mogućnost nadogradnje modulima, sensorima i tzv. „Shields-ovima“ može obavljati različite funkcije te postaje dostupna eksperimentalna oprema za mjerenje[4] i druge tehničko tehnološke svrhe. No najveća dobrobit platforme je integracija tehničkih i prirodnih znanosti u jedinstvenu cjelinu danas objedinjeni pod pojmom STEM (*Science, Tehnology, Engineering, Mathematics*)[4]. Rad s platformom uključuje elemente: programiranja, elektrotehnike, matematike, fizike i strojarstva.

S obzirom na konstruktivistički cilj učenja, odnosno nužnost sudjelovanja učenika u aktivnosti učenja. Učitelji ukazuju da platforma Arduino potiče aktivno učenje što smatraju iznimno važnim korakom prema samostalnom učenju. Uz to, dodaju da u aktivnom istraživanju

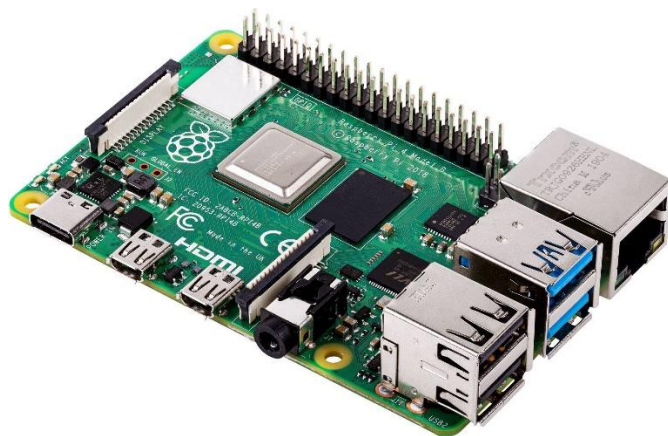
primjenjuju znanja u realnom okruženju čime se dodaju da u aktivnom istraživanju primjenjuju znanja u realnom okruženju čime se ostvaruje dobrobit značenja onoga što se istražuje i uči. Ukazuju na mogućnost rješavanja problema uz uvjet, poznavanje činjenica o pojavi koja se istražuje i temeljna znanja iz većine STEM discipline[4]. Što se tiče izrade i dizajna novih materijala ukazuju da je platforma dizajnirana s ciljem modifikacije, poboljšanja postojećih rješenja i realizacije novih ideja. Ipak, smatraju da je za kreiranje autentičnih projekta nužno veće iskustvo u radu s platformom [4].

2.2. Klasifikacija razvojnih platformi

Pri klasifikaciji spomenuti ćemo Raspberry Pi i Arduino platforme kao dvije različito izvedene razvojne platforme i opisati osnovne značajke. U protekla dva desetljeća razvijene su mnoge razvojne platforme i sustavi koji upravljaju spomenutim razvojnim platformama. Razvoj mikrokontrolera počinje proizvodnjom mikrokontrolera za izradu kalkulatora. Kasnije se javlja ideja o programibilnim mikro kontrolerima koji bi omogućili pojedincima razvoj vlastitih projekata pomoću javno dostupnih „Open Source“ koda. Danas je na tržištu dostupan široki spektar razvojnim platformi, najpopularniji od svih razvojnih platformi su dvije: Arduino i Raspberry Pi. Stoga će u nastavku biti opširnije obrađene ove dvije razvojne platforme.

2.2.1. Raspberry Pi

Raspberry Pi je pristupačno, jeftino računalo izrazito malih dimenzija koje se funkcionalnostima nikako ne razlikuje od standardnog stolnog računala. Za prosječnog računalnog korisnika od standardnog stolnog računala distancira se ponajprije dizajnom, naime sve komponente Raspberry Pi računala smještene su na jednu tiskanu pločicu za razliku od stolnog računala gdje su komponente kao što su procesor, grafička kartica ili pak zvučna kartica izvedene individualno, a računalo kao cjelinu čine kada se sve komponente povežu preko matične ploče [3]. Treba naglasiti da se Raspberry Pi računalo prodaje bez dodatne opreme kao što je kućište ili pak napajanje. Korisniku se ostavlja odabir što sve želi povezati sa računalom pa je sukladno tome primoran nabaviti odnosno kupiti dodatnu opremu [3]. Za programiranje koristi programske jezike Scratch i Python.



Slika 2.1. Raspberry Pi

2.2.2. Prednosti Raspberry Pi razvojne platforme

Raspberry Pi je lagano i prijenosno računalo koje se može spajati sa drugim uređajima i perifernim modulima što ga čini svestranim alatom za razne zadaće. S obzirom na veliki broj GPIO pinova može podržati više senzora, modula i ekrana [15]. U odnosu na ostale mikro kontrolere ima jači i brži procesor, što znači da se može koristiti i za pokretanje manje internetske stranice. Podržava Python programski jezik koji je jednostavniji za razumijevanje i programiranje odnosu na ostale programske jezike [15].

2.2.3. Nedostaci Raspberry Pi razvojne platforme

Raspberry Pi nema ugrađenu unutarnju memoriju zbog čega je potrebno dodatno ugraditi micro SD karticu koja će služiti kao memorija. Negativna strana SD kartica je brzina, što za uzvrat uzrokuje smanjene performansi mikrokontrolera. Originalni Mikro kontroleri nemaju ugrađene sustave hlađenja, to za posljedicu ima pregrijavanje procesora [16].

2.3. Arduino

Arduino sustavi su sustavi, sklopovi ili platforma tzv. otvorenog koda („open-source“) koji služe za kreiranje različitih elektroničkih projekata [17]. Za programiranje koristi vlastiti Arduino jezik, baziran na programskom jeziku Processing [1]. Nadalje, na Arduino platformi se nalaze pinovi (konektori) za napajanje modula, koji upravljaju različitim elektroničkim sklopovima (3.3 V i 5 V), GND (uzemljenje), oin za resetiranje te pinovi za ulazno-izlazne analogno digitalne signale. Analogni ulazi, označeni na platformama oznaka A0 do A5, mogu se koristiti za mjerenje

napona (0V-5V) koji je spojen na njih i pomoću tog podatka upravljati drugim sklopovljem [17]. Sa suprotne strane analognih pinova nalaze se digitalni pinovi, koji se na Arduino platformi označeni oznakama 0 – 13. Ovi pinovi se mogu koristiti kao ulazni i izlazni priključci [17]. Mogućnost za pulsno-širinsku modulaciju (eng. Pulse-Width Modulation, PMW) ima na 6 pinova, odnosno izlaza, a njihova frekvencija je 976.56 Hz i 490.20 Hz. Svaki izlaz može podnijeti najvišu struju od 40 mA, a sveukupno izlazi ne smiju prijeći 200 mA. [1] Po preporuci proizvođača ulazni napon potrebno je održavati između 7 V i 12 V. Postoji više različitih inačica razvojnih pločica koje se najviše razlikuju po broju pinova ulaza i izlaza, brzini i broju mogućih senzora. Najpoznatiji su Arduino Uno, Arduino Nano i Arduino Mega. Važno je napomenuti da se pomoću Arduina Mega mogu složiti i neki ozbiljniji projekti, poput upravljanja pametnim kućama ili manjim proizvodnim linijama [17]. Arduino Mega je posebno dizajniran za projekte koji zahtijevaju složene sklopove i više memorijskog prostora. Većina elektroničkih projekata može se izvesti prilično dobro pomoću drugih ploča dostupnih na tržištu što Arduino Mega čini neuobičajenim za obične projekte. Međutim neke projekte poput izrade 3D pisača ili upravljanja više od jednog motora, zbog njegove sposobnosti pohranjivanja više instrukcija u memoriju koda i broj I/O digitalnih i analognih pinova [17]. Za programiranje koristi vlastiti Arduino jezik. Arduino jezik je pisan u programskom jeziku C++ sa dodatnim specijalnim metodama i funkcijama. Arduino IDE (*Integrated Development Environment*) je glavni program za uređivanje teksta koji se koristi za Arduino programiranje [19].

2.3.1. Prednosti Arduino razvojne platforme

Arduino razvojna platforma koristi pojednostavljenu verziju programskog jezika C/C++ koju svaki početnik može lako naučiti. Važno je napomenuti da ranoj razvojnoj fazi učenika može koristiti i vizualno okruženje kao što su Scratch i Visualino, vizualno programiranje uvelike olakšava uvod u metodu programiranja. Arduino pločice dostupne su u prihvatljivim cijenama i ne zahtjeva dodatnu ulaganju po pitanju dodatnih komponenti jer se mnogi projekti mogu izvršiti koristeći se samo jednom Arduino razvojnom pločicom. Arduino ima zajednicu koja već ima veliki broj razrađenih projekata od jednostavnijih kao što su kontrola ventilatora pa sve do pametnih kuća. Ima i podršku na više platformi za razliku od ostalih razvojnih platformi kao što su Microsoft windows, Linux ili macOS. Na internetu su također dostupne datoteke sa programiranim

senzorima, motorima, modulima i sličnim komponentama što uvelike olakšava izradu željenog sustava.

2.3.2. Nedostaci Arduino razvojnih platformi

Neki od nedostataka Arduino razvojnih platformi svakako je nemogućnosti izvršavanja više zadataka odjednom kao i nedostatak mogućnosti optimizacije performansi. U odnosu na ostale razvojne platforme Arduino razvojna platforma ima znatno manje unutarnje memorije. Premda podržava jezik C/C++ ostale jezike kao što su Python, Java ili JavaScript ne podržava, iako su ti programi daleko pristupačniji i lakši za upotrebu.

3. Robotska ruka

Robotska ruka je vrsta mehaničke ruke, u pravilu programibilne, slične funkcijama ljudske ruke. Neki roboti programirani su vjerno izvoditi određene radnje iznova i iznova (ponavljajuće radnje) bez varijacija i s visokim stupnjem točnosti. [5] Ove radnje određene su programiranim rutinama, koje specificiraju pravac, ubrzanje, brzinu, usporavanje i udaljenost niza koordiniranih pokreta. [5] Ruka može biti ukupni zbroj mehanizama ili može biti dio složenijeg robota. Karike takvog manipulatora povezane su zglobovima koji omogućuju rotacijsko gibanje (kao kod zglobnog robota) ili translativno (linearno) gibanje. Možemo reći da karike manipulatora tvore kinematički lanac [23]. Završetak kinematičkog lanca manipulatora naziva se krajnji efektor i analogan je ljudskoj ruci. Međutim, često se zabranjuje pojam „robotska ruka“ kao sinonim za robotski manipulator. Robotske ruke dijele se po izvedbama manipulatora, to su: kartezijanski robot, kolaborativni robot, cilindrični robot, sferični robot, SCARA robot, zglobni robot i paralelni robot. Serijska robotska ruka može se opisati kao lanac karika koja se pokreću zglobovima koje pokreću motori. Krajnji efektor, koji se naziva i robotska ruka, može se pričvrstiti na kraj lanca. Kao i drugi robotski mehanizmi, robotske ruke se obično klasificiraju prema broju stupnjeva slobode. Obično je broj stupnjeva slobode jednak broju zglobova koji pokreću karike robotske. Potrebno je najmanje šest stupnjeva slobode kako bi ruka robota mogla postići proizvoljan položaj i orijentaciju u trodimenzionalnom prostoru. Dodatni stupnjevi slobode dopuštaju promjenu konfiguracije neke karike na ruci, dok ruka robota ostaje u istoj pozi. Inverzna kinematika je matematički postupak za izračunavanje konfiguracije ruke, obično u smislu zglobnih kutova, zadanom željenom pozom ruke robota u trodimenzionalnom prostoru [23]. Krajnji efektor, ili robotska ruka, može se dizajnirati za obavljanje bilo kojeg željenog zadatka kao što je zavarivanje, hvatanje, bojanje, pakiranje, etiketiranje, paletiranje, testiranje, pređenje i ostalo ovisno o namjeni. Na primjer robotske ruke u automobilskoj industriji na montažnim linijama obavljaju razne zadatke kao što su zavarivanje, rotacija i postavljanje dijelova tijekom sastavljanja. U nekim okolnostima, poželjna je bliska emulacija ljudske ruke, kao kod robota dizajniranih za razoružavanje i uklanjanje bombi. Ostali roboti mnogo su fleksibilniji u pogledu objekta na kojem rade ili čak i zadatka koji treba obaviti na samom objektu, a koji će robot možda i sam morati identificirati. [5] Za preciznije navođenje roboti često sadrže podsustav strojnog oblika, koji

djeluje kao njihov vizualni senzor, povezan s moćnim računalima ili kontrolorom. [5] Umjetna inteligencija postaje sve važniji faktor modernog industrijskog robota. [5]

3.1. Kartezijanski robot (linearni robot)

To je robot (4.6.) čija ruka ima tri zgloba, čije osi koincidiraju s kartezijanskim koordinatnim sustavom, odnosno kreću se pravocrtno, bez rotacije i nalaze se pod pravim kutom jedna prema drugoj. Tri klizna zgloba pomiču robot prema gore, dolje i naprijed nazad. Ima visoku pouzdanost i preciznost pri radu u trodimenzionalnom prostoru. Kao sustav koordinata robota, idealan je za slaganje kutija i horizontalno kretanje. Mogu se izvesti od bilo kojeg linearnog aktuatora sa nizom pogonskih mehanizama (linearnog motora, pneumatskog aktuatora, remena i slično). Jednostavni su za programiranje zbog samo tri osi i imaju visoki stupanj preciznosti. Koriste se kao: 3D printeri, glodalice i crtači.



Slika 3.1. Kartezijanski robot

3.2. Kolaborativni robot

Kobot ili kolaborativni robot (slika 4.7.), je robot namijenjen za izravnu interakciju čovjeka i robota unutar zajedničkog prostora ili tamo gdje su ljudi i roboti u neposrednoj blizini. Primjena kobota su u suprotnosti sa tradicionalnim primjenama industrijski robota u kojima su roboti izolirani od ljudskog kontakta. Sigurnost kobota može se oslanjati na lagane konstrukcijske

materijale, zaobljene rubove i inherentna ograničenja brzine i sile pomoću senzora i softvera koji osiguravaju sigurno ponašanje [20]. Zahvaljujući značajka sigurnog dizajna kolaborativni roboti mogu izravno komunicirati sa ljudima. Međunarodna federacija robotike(IFR), globalna industrijska udruga proizvođača robota i nacionalnih udruga robota: industrijske robote koji se koriste u automatizaciji u industrijskom okruženju i servisne robote za kućnu i profesionalnu upotrebu. Uslužni roboti mogu se također smatrati kobotima jer su namijenjeni za rad zajedno s ljudima. Industrijski roboti tradicionalno su radili odvojeno od ljudi iza ograda ili drugih barijera, ali koboti uklanjaju tu odvojenost. Koboti imaju razne namjene: informacijski roboti, uslužni roboti, logistički roboti koji transportiraju materijale i industrijske robote koji automatiziraju neergonomske poslove čovjeka. IFR(*International Federation of Robotics*) definira četiri razine suradnje između robota i ljudskih radnika [20]:

1. Suživot: čovjek i robot rade jedan pored drugog bez ograde, ali bez zajedničkog radnog prostora,
2. Sekvencijalna suradnja: čovjek i robot aktivni su u zajedničkom radnom prostoru, ali su njihovi pokreti sekvencijalni (ne rade u isto vrijeme),
3. Suradnja: robot i čovjek rade na istom dijelu u isto vrijeme, a oboje su u pokretu,
4. Odgovorna suradnja: robot u stvarnom vremenu reagira na kretanje ljudskog radnika (ima veliku važnost jer sprječava ozljede na radu i omogućava blisku suradnju robota i čovjeka) [20].

U većini današnjih industrijskih primjena kobota, kobot i ljudski radnik dijele isti prostor, ali izvršavaju zadatke neovisno ili uzastopno. Suradnja ili odgovorna suradnja trenutno je manje uobičajena [20].



Slika 3.2. Kolaborativni robot

3.3. Cilindrični robot

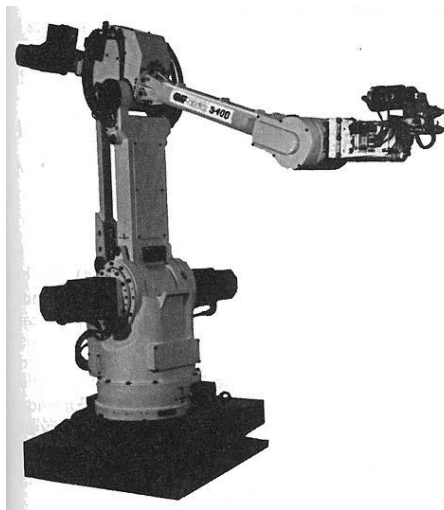
Cilindrični roboti (slika 4.8.) su roboti čije osi tvore cilindrični koordinatni sustav. Koriste se za: sastavljanje, rukovanje alatnim strojevima, točkasto zavarivanje, rukovanje strojevima za tlačno lijevanje i sličnim zadaćama u industriji. Glavna ruka može se kretati gore-dolje. Robot može izvesti ovaj pokret produživši cilindar ugrađen u ruku. Kod većine cilindričnih robota gibanje prema gore i dolje osigurava se pneumatskim cilindrom, a zakretanje se obično osigurava motorom i zupčanicima[6]. Roboti ove vrste obično se mogu kretati duž Z i Y osi i rotirati oko Z osi, stoga ovakvo kretanje čini cilindrični koordinatni sustav. Ovakvi tipovi robota sve se rjeđe pojavljuju, ali još uvijek postoje područja gdje su i danas korisni, npr. Montaže ili točkasto zavarivanje[6].



Slika 3.3. Cilindrični robot

3.4. Sferični robot

Sferični roboti (slika 4.9.) ili polarni roboti slični su cilindričnim robotima, ali umjesto jednog rotacijskog zgloba i dva linearna spoja, imaju dva rotacijska zgloba i jedan linearni spoj. Ovi se roboti koriste u nekim najosnovnijim robotskim aplikacijama, poput bojanja zavarivanja i sastavljanja[6]. Danas ih na tržištu ima sve manje. Međutim, sferični robot je onaj koji je sve pokrenuo. Bez sferne robotske tehnologije nikada ne bi bilo prvog industrijskog robota. Ovaj je robot stvorio put novim tehnologijama poput današnjih šestero osnih robota. Iako imaju samo tri osi, sferični robotski sustavi poput Unimate-a bili su u mogućnosti znatno ubrzati proizvodnju[6].



Slika 3.4. Sferični robot

3.5. SCARA roboti

SCARA (*Selective Compliance Assembly Robot Arm*) (slika 4.10.) odnosno „Selektivna usklađenost artikulirane robotske ruke“. Popustljiv u X i Y osi, nepomičan u osi Z i ima dva rotacijska i jedan translacijski zglob. U pravilu brži u odnosu na kartezijske robote zbog čega su i skuplji. Dolaze sa već instaliranim softverom koji je transparentan za korištenje. S obzirom da je artikuliran (može se potpuno sklopiti nakon ekstenzije), idealan je za prijenos tereta u skućenim prostorima. Dva međusobno paralelna rotacijska zgloba u vertikalnim osima montirana su na postolju baze tako da se oba segmenta kreću u horizontalnoj ravnini[6]. Na kraju drugog segmenta

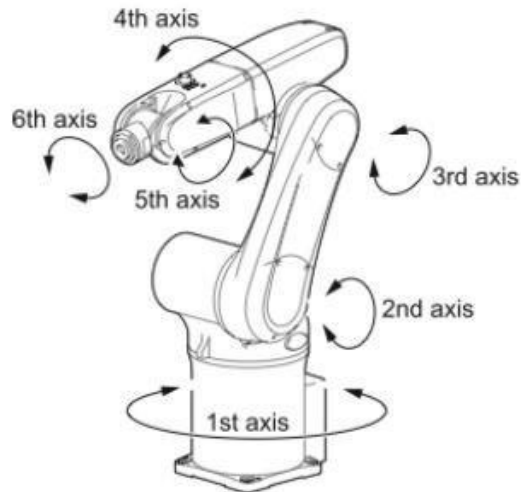
nalazi se translatorni zglob čija os je također vertikalna. Traslatorni zglob najčešće ima samo jedna stupanj slobode i to okretanje oko vertikalne osi [6].



Slika 3.5. SCARA robot

3.6. Zglobni robot

Zglobni robot (slika 4.11.) je robot sa rotirajućim zglobovima koji imaju raspon od dva pa više i od deset zglobova (najčešće imaju šest) . Mogu biti artikulirani ili ne artikulirani. Svaka dodani zglob (os) omogućuje dodatnu slobodu kretanja robota a time i izvršavanje određenih zadaća. U pravilu su zglobovi raspoređeni u lanac kako bi svaki zglob mogao poduprijeti idući zglob [21]. Najčešće ih pokreću servo motori ali postoje i inačice sa elektromotorima. Najpopularniji tip robota primarno zbog slobode kretanja, velike preciznosti i prilagodljivosti raznim zadaća u industriji. Te primjene uključuju: prijenos dijelova, montaža, lijepljenje, elektrolučno zavarivanje, rukovanje materijalom, odabir i postavljanje, pakiranje, utovar u stroj, paletiranje i mnoge druge [21].



Slika 3.6. Zglobni robot (3)

Opis radnji osi robota: Os 1. rotira cijeli robot oko osi 1. Os 2 pomiče robotsku ruku pomoću prvog kraka, naprijed i nazad. Drugi krak se podiže i spušta pomoću osi 3. a rotira se pomoću oko osi 4. Efektor se podiže i spušta oko osi 5. a os šest spušta i diže efektor.

3.7. Paralelni robot

Paralelni (delta) (slika 4.12.) robot konstrukcijski se sastoji od 3 ili više kinematičkih ruku spojenih univerzalnim zglobovima u bazi robota. Ključna značajka sustava je korištenje paralelograma u krakovima, koji održavaju orijentaciju krajnjeg efektor. Ključni koncept delta robota je uporaba paralelograma koji ograničavaju kretanje krajnje platforme na čistu translaciju, odnosno samo kretanje u smjeru X, Y i Z bez rotacije. Baza robota nalazi se iznad radnog prostora i na njoj su smješteni svi aktuatori. Pokretanje se može izvesti linearnim ili rotacijskim pogonima, sa ili bez redukcije (izravni pogon). S obzirom da su aktuatori smješteni u bazu robota to omogućuje izradu ruke robota od laganih kompozitnih materijala. S obzirom na smanjenu težinu to im omogućuje veće brzine. Delta roboti su veoma popularni u tvornicama gdje se koriste za princip rada „pokupi i spremi“ jer mogu biti veoma brzi čak i do tristo skupljanja u minuti.



Slika 3.7. Paralelni robot

3.8. Servo motori

Servo motor je rotacijski ili linearni aktuator koji omogućuje preciznu kontrolu kutnog ili linearnog položaja, brzine i ubrzanja[1]. Sastoji se od odgovarajućeg motora spojenog sa senzorom za povratnu informaciju o položaju. Također zahtjeva relativno sofisticiran upravljač, često namjenski modul dizajniran posebno za korištenje sa servomotorima. Servo motori nisu posebna klasa motora, iako se izraz servo motor koristi za označavanje motora prikladnog za upotrebu u sustavu upravljanja zatvorenom petljom. Servomotori se koriste u poljima CNC strojeva i automatizirane proizvodnje[1]. Servo motor je servo mehanizam zatvorene petlje koji koristi povratnu informaciju o položaju za upravljanje svojim gibanjem i konačnim položajem. Ulaz za njegovu kontrolu je signal(bilo analogni ili digitalni) koji predstavlja položaj naređen za izlaznu osovinu. Motor je uparen s nekom vrstom enkodera položaja za pružanje povratne informacije o položaju i brzini. U najjednostavnijem slučaju mjeri se samo položaj. Izmjereni položaj izlaza uspoređuje se s položajem naredbe, vanjskim ulazom u upravljač. Ako se izlazni položaj razlikuje od potrebnog, generira se signal pogreške koji zatim uzrokuje rotaciju motora u bilo kojem smjeru, koliko je potrebno da se izlazna osovina dovede u odgovarajući položaj. Kako se položaji približavaju, signal pogreške smanjuje se na nulu i motor se zaustavlja.

4. Projekt – Razvoj robotske ruke

Cilj je edukacijski obraditi projekt, njegove komponente, zahtjeve i mogućnosti primjene u edukaciji inženjera. Zahtjev projekta je razumjeti ulogu individualnih komponenti u upravljanju robotskom rukom, spajanje tih komponentni, programiranje i upravljanje robotskom rukom. Za početak će biti navedene potrebne komponente za slaganje sustava robotske ruke. Potom ćemo ih detaljno objasniti i prikazati programski kod koji se koristi za programiranje individualnih komponenti sustava i objasniti način spajanja. Robot je zglobova ruka koja se kreće po 3 osi (pokreće ga 6 servo motora). Robotska ruka će biti programirana da bilježi svoje položaje. Nakon snimanja, slijed položaja može se automatski reproducirati i ruka animirati. Položaji ostaju pohranjeni i kada se struja prekine. Cijelim sustavom se upravlja putem pametnog telefona spojenim pomoću Bluetooth aplikacije.

4.1. Komponente robotske ruke

Automatizirana robotska ruka koristi softverske i automatizirane komponente, koje su lako dostupne na tržištu po pristupačnim cijenama. Nakon sastavljanja robotskom rukom upravljati će Arduino Uno, pomoću softvera Arduino IDE. Robotska ruka ima kostur od aluminija koju pokreće 6 servo motora. Servo motore spaja i regulira Adafruit servo driver. Sustav je međusobno spojen PC jumper vodovima. Podaci o programu se spremaju u AT24C256 EEPROM sustav. Sve komponente izuzev Arduino Uno-a su spojeni pomoću probne pločice. Komponente koje su korištene u ovom projektu će se opisati u nastavku, a uključuju:

Komponente hardvera:

- HC-06 Bluetooth Modul,
- AT24C256 Serial EEPROM I2C Interface EEPROM Dana Storage Module for Arduino,
- Adafruit PCA9685 16-channel Servo Driver,
- Arduino UNO,
- Robotska ruka,
- LCD 16x2 ekran
- 10 Pc. Jumper Wire kit, 5cm dugi,
- Testna pločica.

Za realizaciju softverskog dijela projekta koristiti će se Arduino razvojno sučelje (Arduino IDE).

4.2. HC-06 Bluetooth Modul

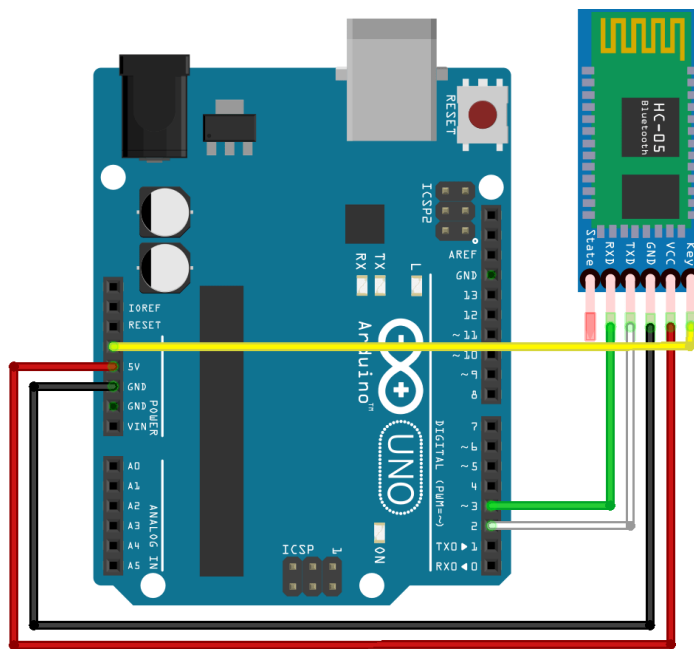
HC-06 (slika 4.1.) se koristi se za mnoge aplikacije poput bežičnih slušalica, kontrolera za igre, bežičnog miša bežične tipkovnice i mnogih drugih korisničkih aplikacija. Ima domet do 100 metara što ovisi i o odašiljaču i prijammniku atmosferi, geografskim i urbanim uvjetima. Koristi radio tehnologiju proširenog spektra frekvencijskog skakanja za slanje podatak putem zraka. Komunicira s mikro kontrolerom putem serijskog priključka(USART). Može se koristiti u podređenoj ili glavnoj konfiguraciji. Primjer spajanja Bluetooth modula na Arduino UNO prikazan je na slici 4.2.



Slika 4.1. Bluetooth modul

Tipka/EN: Koristi se za postavljanje Bluetooth modula u način rada AT naredbi. Ako je Pin Key/EN postavljen na visoku vrijednost tada će ovaj modul raditi u naredbenom načinu rada. U suprotnom je u naredbenom načinu rada. Pin VCC služi za spajanje napajanja. Pin za uzemljenje modula je GND. Pin TXD služi za prijenos serijskih podataka (bežično primljeni podaci preko Bluetooth modula koji se prenose serijski na TXD pinu). Pin RXD zaprima podatke serijski(primljeni podaci će se bežično prenositi Bluetooth modulom. Indikator stanja govori je li modul spojen ili ne. HC-06 ima crvenu LED diodu koja označava status veze, bez obzira je li Bluetooth povezan ili ne. Prije spajanja na HC-05 modul crvena LED dioda treperi kontinuirano

na periodički način. Kada se poveže s bilo kojim drugim uređajem, njegovo treptanje usporava se na dvije sekunde.



Slika 4.2. Način spajanja Bluetooth modula

4.3. Memorijski modul (AT24C256 EEPROM)

Memorijski modul (AT24C256) (slika 4.3.) pruža 131,072/262,144 bita serijske elektronički izbrisive i programibilne memorije samo za čitanje (EEPROM) organizirane kao 16,384/32,768 riječi od po 8 bita. Kaskadna značajka uređaja omogućuje do 4 uređaja da dijele zajedničku dvožilnu sabirnicu. Uređaj je optimiziran za korištenje u mnogim industrijskim i komercijalnim primjenama gdje je bitan rad niske snage i napona. Uređaji su dostupni u 8-vodnom JEDEC PDIP, 8-vodnom JEDEC SOIC, 8-vodnom EIAJ SOIC, 8-vodnom MAP, 8-vodnom TSSOP, 8-vodnom SOIC Array paketu i 8-loptičkom dBGA2 paketu. Uz to cijela paleta proizvoda dostupna je u verzijama od 2,7 do 5,5 volti i 1,8 do 3,6 volti. Ova će nam memorija koristiti za spremanje unaprijed isprogramirane radnje robotske ruke. Značajke memorijskog modula: rad na niskom i standardnom naponu, dvožično serijsko sučelje, Schmittov okidač, filtrirani ulazi za suzbijanje smetnji, protokol za dvosmjerni prijenos podataka, pin za zaštitu od

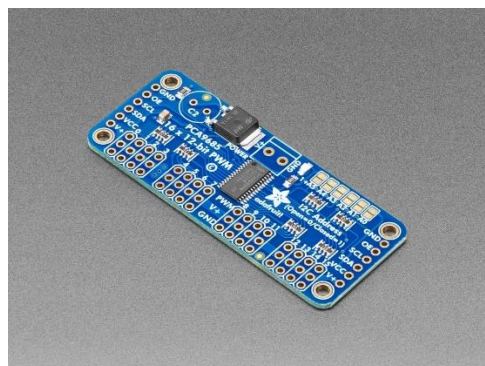
pisanja za hardversku i softversku zaštitu podataka, način pisanja stranica od 64 bajta, samo tempirani ciklus pisanja i visoka pouzdanost i izdržljivost.



Slika 4.3. EEPROM memorija

4.4. Proširenje Adafruit PCA9685

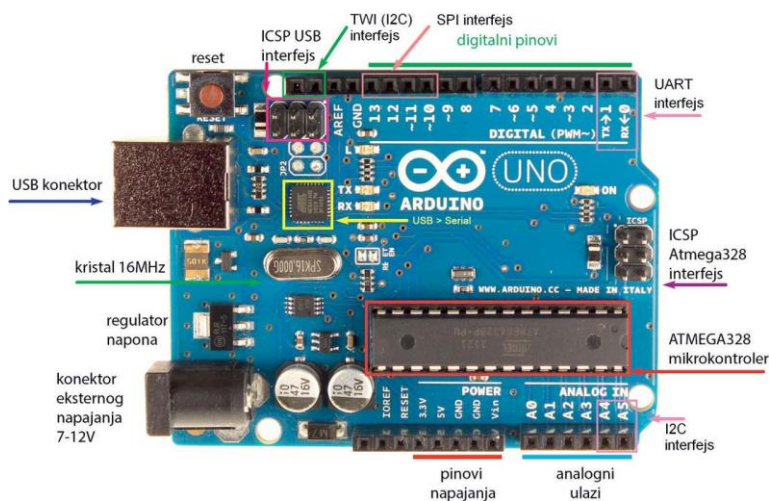
Proširenje Adafruit PCA 9685 (slika 4.4.) je 12C-kontrolirani PWM driver s ugrađenim satom, što znači da u odnosu na TLC5940, ne mora mu se neprestano slati signal. Koristi izvor napona od 5 volti stoga njime možemo upravljati mikro kontrolerom od 3,3 volta, s logičkim povlačenjem od 3,3 volta i još uvijek sigurno spuštati od izlaza od 5,5 volta. Ima 6 ulazni pinova za odabir adrese tako da možete spojiti do 62 pina na jednu i2c sabirnicu, odnosno sve ukupno 992 izlaza. Podesiva frekvencija PWM do oko 1,6 kHz, 12-bitna razlučivost za svaki izlaz.



Slika 4.4. Servo modul

4.5. Arduino UNO

Arduino UNO (slika 4.5.) je mikro kontrolorska ploča bazirana na ATmega328P. Ima 14 digitalnih ulazno/izlaznih pinova (od kojih se 6 može koristiti kao PWM izlazi), 6 analognih ulaza, keramički rezonator od 16 MHz, USB priključak, utičnicu za napajanje, ICSP zaglavlje i gumb za resetiranje. Sadrži sve što je potrebno za podršku mikro kontrolera, jednostavno se spaja na računalo USB kabelom ili ga napajate adapterom ili baterijom da biste započeli.

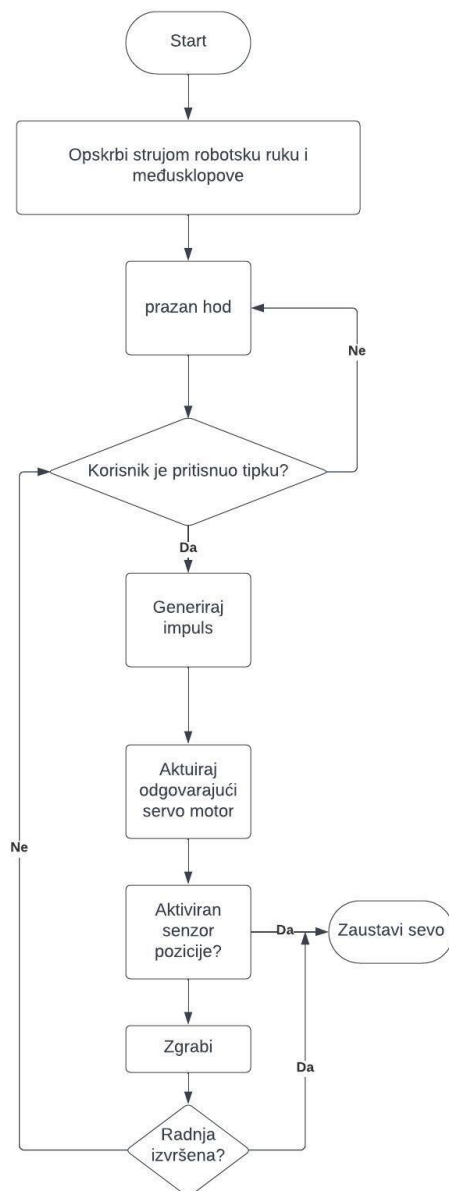


Slika 4.5. Arduino UNO

4.6. Programiranje komponenti robotske ruke

Program koristi tri datoteke: glavni program (slika 4.7. i slika 4.8.) i dvije funkcijske datoteke. Korištenjem funkcija izbjegava se ponavljanje koda i olakšava čitanje. Glavna petlja odnosi se na prijem znaka s Bluetootha. Ovisno o primljenom liku, izvršava se radnja. Od „A“ do „L“ ručno pomičemo servo motor s lijevom ili desnom rotacijom (funkcija „servowrite“). Sada prelazimo na petlju „while“. Servo motor se okreće dok ne dobije znak „S“. „S“ se šalje kada se otpusti gumb na pametnom telefonu. Njime također kontroliramo kurs servo motora kao bi spriječili da udari u potporni krak. Zaprimanje znakova „M“ do „9“ dovodi do posebnih operacija. „M“ i „N“ djeluju na pokazivač sekvence povećanjem ili smanjenjem za jedan. „Y“ prikazuje kut 6 servo motora i vrijednost dva pokazivača sekvence. „Z“ pomiče ruku iz trenutnog položaja u

početni položaj. „1“ do „4“ pomiče ruku iz njezinog trenutnog položaja u položaj koji odgovara nizovima od 1 do 4. „8“ pomiče ruku iz trenutnog položaja u položaj koji odgovara odabranom nizu (slijed odabran pomoću „M“ i „N“). „7“ prikazuje prvih 100 bajtova EEPROM-a na serijskom priključku. „9“ bilježi trenutni položaj ruke (slijed odabran pomoću „M“ i „N“). „6“ započinje slijed pokreta definiranih u tablici „prog“ (funkcija scenarij) i odgovara tom već programiranom scenariju [22]. Na slici 4.6. je prikazan dijagram toka.



Slika 4.6. Dijagram toka

```

1 // Glavni program - upravljanje robotom
2 // LCD ekran =====
3 #include <LiquidCrystal_PCF8574.h>
4 #include <Wire.h>
5 LiquidCrystal_PCF8574 lcd(0x38); // I2C
6 int lcdpres = 0; // stavlja vrijednost 0 ako se LCD ekran ne koristi
7 // EEPROM =====
8 #define EEPROM_I2C_ADDRESS 0x50 // I2C
9 int eprdata[6]; // podaci čitani iz EEPROM memorije
10 int epractmax[2]; // pozicija pokazivača u odnosu na EEPROM memoriju
11 // BLUETOOTH =====
12 #include "SoftwareSerial.h"
13 SoftwareSerial bluetooth(2, 3); // RX, TX
14 int bluerec ; // varijabla za prijam Bluetooth-a
15 // SERVO MOTORI =====
16 #include <Adafruit_PWMServoDriver.h>
17 Adafruit_PWMServoDriver pca= Adafruit_PWMServoDriver(0x40); // I2C
18 #define nbPCAServo 6
19 int MIN_IMP [nbPCAServo] ={500, 500, 500, 500, 500, 500};
20 int MAX_IMP [nbPCAServo] ={2500, 2500, 2500, 2500, 2500, 2500};
21 int MIN_ANG [nbPCAServo] ={0, 0, 0, 0, 0, 0} ;
22 int MAX_ANG [nbPCAServo] ={180,180, 180, 180, 180, 180} ; // Kutevi servo motora
23 int tabanglim[2][6] = {{0,43,65,0,10,80}, {180,177,180,140,175,116}}; // Limitacija servo udarca
24 int servodelay(15); // Odgoda pokretanja servo motora
25 int servo; // Broj trenutnog servo motora (od 0 to 5)
26 int homepos[6] = {93, 124, 115, 118, 83, 80}; // Početna pozicija robotske ruke koja se sprema u EEPROM memoriju
27 int tabserpos[6] = {90,90,90,90,90,90}; // naredba za spremanje trenutne pozicije servo motora
28 String rot; // trenutni smjr rotacije (lijevo ili desno)
29 // Ostale varijable =====
30 int prog[2][10] = {{2,5,6,3,7,8,4,9,11,0}, {0,1,0,0,1,0,0,1,0,0}}; // jedan od mogućih kretanja kao primjer
31
32 //
33 // == POSTAVLJANJE =====
34 void setup() {
35 // inicijalizacija =====
36 Serial.begin(9600);
37 delay(500);
38 // LCD inicijalizacija =====
39 Wire.begin();
40 Wire.beginTransmission(0x38);
41 if (lcdpres == 1) {lcd.begin(16, 2); lcdw(0,0,0, " "); lcdw(1,0,0, " READY TO WORK"); }
42 // Inicijalizacija Bluetooth-a =====
43 bluetooth.begin(9600);
44 // Inicijalizacija servo motora =====
45 pca.begin();
46 pca.setPWMPFreq(60); // Analogni servo motori rade na frekvenciji od 60 Hz
47 delay(500);
48 ..

```

Slika 4.7. glavni dio programa (prvi dio)(22)

```

48 // EEPROM inicijalizacija =====
49 // EEPROM struktura : prva dva bita sadrže trenutni pokazivač
50 // 6 bitova predstavlja 6 servo motora
51
52 readeprom(0,2); epractmax[0] = eprdata[0]; epractmax[1] = eprdata[1]; // pos 0,1 of eeprom indexes
53 delay(100);
54 writeeprom(2,homepos,6); // unos početne pozicije robota
55 delay(100);
56 servomove(0,0); // stavi probot u početni položaj
57 // završavanje inicijalizacije EEPROM memorije
58 Serial.println(" READY TO WORK");
59 }
60 // == PETLJA =====
61 void loop() {
62 delay(100);
63 if (bluetooth.available()) {
64   bluerec = bluetooth.read();
65   switch(bluerec)
66   {
67     case 'A': servo = 0; rot = 'L'; break;
68     case 'B': servo = 0; rot = 'R'; break;
69     case 'C': servo = 1; rot = 'L'; break;
70     case 'D': servo = 1; rot = 'R'; break;
71     case 'E': servo = 2; rot = 'L'; break;
72     case 'F': servo = 2; rot = 'R'; break;
73     case 'G': servo = 3; rot = 'L'; break;
74     case 'H': servo = 3; rot = 'R'; break;
75     case 'I': servo = 4; rot = 'L'; break;
76     case 'J': servo = 4; rot = 'R'; break;
77     case 'K': servo = 5; rot = 'L'; break;
78     case 'L': servo = 5; rot = 'R'; break;
79     case 'M': if (epractmax[0] <= (epractmax[1])) {epractmax[0] = epractmax[0] + 1; listpos(); } rot = ' '; break; // podigni EEPROM indeks
80     case 'N': if (epractmax[0] > 0) {epractmax[0] = epractmax[0] - 1; listpos(); } rot = ' '; break; // smanji EEPROM indeks
81     case 'Y': listpos(); rot = ' '; break; // prikaži poziciju robota
82     case 'Z': servomove(0,0); rot = ' '; break; // pomakni robota u početnu poziciju
83     case '1': servomove(1,0); rot = ' '; break; // pomakni robota u zadanu poziciju
84     case '2': servomove(2,0); rot = ' '; break; // pomakni robota u zadanu poziciju
85     case '3': servomove(3,0); rot = ' '; break; // pomakni robota u zadanu poziciju
86     case '4': servomove(4,0); rot = ' '; break; // pomakni robota u zadanu poziciju
87     case '6': scenario(); rot = ' '; break; // pomakni robota u zadanu poziciju
88     case '7': eeprom_read_test(); rot = ' '; break; // prikaži sadržaj EEPROM memorije
89     case '8': servomove(epractmax[0],0); rot = ' '; break; //pomakni robota u početnu poziciju
90     case '9': writeprom(epractmax[0]); rot = ' '; break; // zapiši poziciju robota u EEPROM memoriju
91     default: rot = ' '; break;
92   }
93   while((bluerec != 'S') && (rot != " "))
94   {
95     if ((rot == "L") && (tabserpos[servo] > tabanglim[0][servo])) { tabserpos[servo] = tabserpos[servo] - 1; } // smanji poziciju servo motora
96     if ((rot == "R") && (tabserpos[servo] < tabanglim[1][servo])) { tabserpos[servo] = tabserpos[servo] + 1; } // podigni poziciju servo motora
97     servowrite(servo, tabserpos[servo]);
98     delay(100);
99     bluerec = bluetooth.read();
100   }
101 }
102 } // kraj petlje
103

```

Slika 4.8. glavni dio programa (drugi dio) [22]

4.6.1. Deklaracija LCD zaslona

LCD zaslon se koristi za prikaz pozicije robotske ruke, trenutni status robota. Koristi se postavka funkcije „lcdf“. Pomoću nje inicijaliziramo LCD ekran za postavljanje, upis teksta ili čišćenje prikaza ekrana (slika 4.9.). Funkcija se priziva prilikom pokretanja robotske ruke i postavlja parametre prikaza ovisno o unesenim podacima (slika 4.10. i slika 4.11.). Ako se zaslon ne koristi, možemo ostaviti deklaraciju. Varijabla „Icdpres“ biti će inicijalizirana na 0. U tom slučaju će informacije biti prikazane na računalo [22].

```

1 // Funkcije za LCD kontrole =====
2 void lcdw (int lcdf, int lcdl, int lcdc, String lcdt) {
3
4 // Setting of the function =====
5 //   lcdf : 0=inicijaliziraj LCD za void postavljanje/1=upis teksta/2=ocisti
      display,
6 //   lcdl : za lcdf=1 prikazi redni broj 0-1,
7 //   lcdc : za lcdf=1 prikazi broj u stupcu 0-15,
8 //   lcdt : za lcdf=1 string za prikazati,
9 // Kraj postavki za funkcije =====
10
11 // Pokretanje funkcija =====
12 switch (lcdf) {
13     case 0:
14         lcd.setBacklight(255);
15         lcd.home();
16         lcd.clear();
17         delay(100);
18     break;
19     case 1:
20         lcd.setBacklight(255);
21         lcd.setCursor(lcdc, lcdl);           // postavljanje
            početka za upis
22         lcd.print(lcdt);                   // upis
            infomacija
23     break;
24     case 2:

```

Slika 4.9. Deklaracija LCD zaslona (prvi dio) [22]

```

22     lcd.print(lcdt);                                     // upis infomacija
23     break;
24     case 2:
25         lcd.clear();                                   // ocisti LCD ekran
26         break;
27     }
28
29 // ===== upiši poziciju robota u EEPROM memoriju
30 void writeprom(int writeprom1){
31     int writeprom3[6];
32     writeprom3[0] = tabserpos[0]; writeprom3[1] = tabserpos[1]; writeprom3[2] = tabserpos[2];
33     writeprom3[3] = tabserpos[3]; writeprom3[4] = tabserpos[4]; writeprom3[5] = tabserpos[5];
34     writeeprom((writeprom1*6 + 2),writeprom3,6) ; // upisuje pozocoiju robota u EEPROM memoriju
35     if (epractmax[0] > epractmax[1]) { epractmax[1] = epractmax[0]; }
36     writeeprom(0,epractmax,2); // unosi vrijedjnost pokazivača u EEPROM memoriju
37     if (lcdpres == 1) {
38         lcdw(2,0,0,""); lcdw(1,0,0,"napisi OK"); lcdw(1,1,0,String (epractmax[0]) + "-" +
           String (epractmax[1]));
39     }
40     Serial.println("napisi OK ");
41     listpos();
42 }
43 // ===== eeprom funkcija upisa
44 void writeeprom(int data_addr, int stotab[], int len)
45 {
46     Wire.beginTransmission(EEPROM_I2C_ADDRESS);
47     Wire.write((int)(data_addr >>8));
48     Wire.write((int)(data_addr & 0xFF));
49     for (int i=0;i!=len; i++) { Wire.write(stotab[i]);}
50     Wire.endTransmission();
51     delay(5);
52 }

```

Slika 4.10. Deklaracija LCD zaslona (drugi dio) [22]


```

53 // ===== eeprom funkcija citanja
54 byte readeeprom(int data_addr, int len)
55 {
56     int i=0;
57     Wire.beginTransmission(EEPROM_I2C_ADDRESS);
58     Wire.write((int)(data_addr >>8));
59     Wire.write((int)(data_addr & 0xFF));
60     Wire.endTransmission();
61     Wire.requestFrom(EEPROM_I2C_ADDRESS, len);
62     delay(10);
63     while (Wire.available()){ eprdata[i] = Wire.read(); i=i+1; }
64 }
65 // ===== eeprom citanje i ispis na keranu racunala
66 void eeprom_read_test()
67 {
68     String str;
69     int j, k=0 ,l=0;
70     for(int i=2;i<100;i++)
71     {
72         j = (i-2)/6 ;
73         if (k != j) { k = j; l = 0;}
74         readeeprom(i,1); delay(10) ; l = l+1;
75         str = "Pozicija - Servo : " + String((i-2)/6) + " - " + String(l) + " vrijednost: " + String(eprdata[0]);
76         Serial.println(str) ;
77     }
78     Serial.println("Kraj liste");
79 }
80 void reinitI2CByte()
81 {
82     int data_addr = 0;
83     Wire.beginTransmission(EEPROM_I2C_ADDRESS);
84     Wire.write((int)(data_addr >>8));
85     Wire.write((int)(data_addr & 0xFF));
86     for (int i=0;i < 30; i++) { Wire.write(0);}
87     Wire.endTransmission();
88     delay(5);
89 }
90

```

Slika 4.11. Deklaracija LCD zaslona (treći dio) [22]

4.6.2. Deklaracija EEPROM memorije

Adresa na 12C sabirnici je 0x50. Tablica „eprdata“ koristi se kao među spremnik za razmjenu s memorijom. Sadrži 6 bajtova koji odgovaraju 6 servo uređaja i danom položaju ruke. Hod servo koji ide od 0 do 180 pohranjuje se u bajtu. Stoga je u projektu korišteno 6 bajtova po položaju ruke. Ovih 6 vrijednosti tada će se zvati sekvence i biti će pročitane ili zapisane u memoriju. Koristimo dva pokazivača ili redni broj. Pokazivač za broj trenutne sekvence i pokazivač za broj posljednje sekvence snimljene u Eeprom memoriji. Niz „eptactmax“ sadrži vrijednosti ova dva pokazivača [22].

4.6.3. Bluetooth deklaracija

Modul je spojen na pin 2 i 3 Arduino ploče. Varijabla „bluerec“ sadrži primljenu informaciju (slovo abecede). U ovom projektu „Remote“ aplikacija će poslati samo jedan znak po akciji [22].

4.6.4. Deklaracija servo kontrolera

Za ovaj dio je potrebno najviše opreza. Adresa na 12C sabirnici je 0x40. Program koristi kut rotacije (0 do 180) za pozicioniranje servo motora. Rotaciju provodi PWM kontroler koji koristi impulse čija je širina (500 do 2500 μ s) varira ovisno o kutu. Komunikacija s kontrolerom obavlja se pomoću funkcije „servowrite“ koja će prethodno izvršiti pretvorbu u kut-plus. Za ovu pretvorbu koriste se varijable „MAX i MIN“. Ovisno o servo modelu ove se vrijednosti mogu mijenjati. U našem slučaju servo MG996 ima kut rotacije od 180° stupnjeva (slika 4.12. i slika 4.13.) Zamjenski servo DS3225 ima kut od 270° stupnjeva. Zadržat ćemo rotaciju od 180° jer rotacija od 270° mehanički nije izvediva za ovaj krak. U sklopu se neki servo motori ne mogu okretati preko čitavog raspona od 180° stupnjeva. Oni će biti mehanički blokirani nosačima i početi će se pregrijavati. Stoga je potrebno zaštititi ih ograničavanjem njihovog udara. Tablica „tabanglim“ sadrži granične vrijednosti za rotaciju svakog servo motora (min i max). Tablica „homepos“ sadrži položaj ruke pri uključivanju. Preporuča se vratiti ruku u ovaj položaj prije isključivanja, jer će tada prilikom ponovnog uključivanja biti na tom mjestu. Ako je ruka u drugom položaju, pri uključivanju će se dogoditi nagli pokret koji može oštetiti servo motore [22].

Princip pokreta temelji se na istovremenoj (prividnoj) rotaciji servo motora. Bilo bi jednostavnije rotirati jedan servo za drugim do određene pozicije. Vizualni učinak tada nije gladak, a pokret trza. Program je stoga malo složeniji. Radimo u serijama servo motora. Svaki servo se okreće korak po korak [22].

Servo motori su označeni brojevima od 0 (baza) do 5 (stezaljka). Rotacija se vrši sljedećim redoslijedom:

1. Serija koja se sastoji od servo uređaja 1 i 2 pomiče se prema gore neovisno o nizu.
2. Serija sastavljena od servo uređaja 0, 3 i 4 pomiče se prema vrijednostima sekvence.

3. Serija sastavljena od servo motora 1 i 2 pomiče se prema vrijednosti sekvence.
4. Serija 5 (stezaljka) se pomiče prema vrijednostima sekvence [22].

```

1 // ===== pomicanje robota u poziciju
2 void servomove(int li,int sk){ // pomakni robota u zadanu poziciju
3 // Postavljanje funkcija =====
4 delay(200); // kako bi bili sigurni da je prijašnja radnja izvršena
5 int bb = 75,cc = 110;
6 int ind0=1, ind1=1, ind2, ind3, ind4, ind5 ;
7 if (sk == 1) {ind1 = 0; ind2 = 0;}
8 while (ind1+ind2 >0)
9 {
10 if (tabserpos[1] != bb) { ind1 = 1;
11 if (tabserpos[1] < bb) { tabserpos[1] = tabserpos[1]+1; } if (tabserpos[1] > bb) { tabserpos[1] =
    tabserpos[1]-1; } servowrite(1, tabserpos[1]); }
12 else { ind1 = 0 ;}
13 if (tabserpos[2] != cc) { ind2 = 1;
14 if (tabserpos[2] < cc) { tabserpos[2] = tabserpos[2]+1; } if (tabserpos[2] > cc) { tabserpos[2] =
    tabserpos[2]-1; } servowrite(2, tabserpos[2]); }
15 else { ind2 = 0 ;}
16 } // end while (ind1+ind2 >0)
17
18 if (li > epractmax[1]) { li = epractmax[1] ; }
19 readeeprom((li*6 + 2),6) ; //EEPROM meorija sadrži pozicije servo motora
20 ind0 = 1;
21 while (ind0+ind3+ind4+ind5 >0)
22 {
23 if (tabserpos[0] != eprdata[0]) { ind0 = 1;
24 if (tabserpos[0] < eprdata[0]) { tabserpos[0] = tabserpos[0]+1; } if (tabserpos[0] > eprdata[0]) {
    tabserpos[0] = tabserpos[0]-1; } servowrite(0, tabserpos[0]) ; }
25 else { ind0 = 0 ;}
26 if (tabserpos[3] != eprdata[3]) { ind3 = 1;
27 if (tabserpos[3] > eprdata[3]) { tabserpos[3] = tabserpos[3]-1; } if (tabserpos[3] < eprdata[3]) {
    tabserpos[3] = tabserpos[3]+1; } servowrite(3, tabserpos[3]) ; }
28 else { ind3 = 0 ;}
29 if (tabserpos[4] != eprdata[4]) { ind4 = 1;
30 if (tabserpos[4] < eprdata[4]) { tabserpos[4] = tabserpos[4]+1; } if (tabserpos[4] > eprdata[4]) {
    tabserpos[4] = tabserpos[4]-1; } servowrite(4, tabserpos[4]) ; }
31 else { ind4 = 0 ;}
32 }

```

Slika 4.12. Deklaracija servo kontrolera (prvi dio) [22]

```

32 }
33 ind1 = 1;
34 while (ind1-ind2 >0)
35 {
36- if (tabserpos[1] != eprdata[1]) { ind1 = 1;
37   if (tabserpos[1] < eprdata[1]) { tabserpos[1] = tabserpos[1]+1; } if (tabserpos[1] > eprdata[1]) { tabserpos[1] = tabserpos[1]-1
   ; } servowrite(1, tabserpos[1]) ; }
38   else { ind1 = 0 ; }
39- if (tabserpos[2] != eprdata[2]) { ind2 = 1;
40   if (tabserpos[2] < eprdata[2]) { tabserpos[2] = tabserpos[2]+1; } if (tabserpos[2] > eprdata[2]) { tabserpos[2] = tabserpos[2]-1
   ; } servowrite(2, tabserpos[2]) ; }
41   else { ind2 = 0 ; }
42 }
43 ind5 = 1;
44 while (ind5 >0)
45 {
46- if (tabserpos[5] != eprdata[5]) { ind5 = 1;
47   if (tabserpos[5] < eprdata[5]) { tabserpos[5] = tabserpos[5]-1; } if (tabserpos[5] > eprdata[5]) { tabserpos[5] = tabserpos[5]
   -1; } servowrite(5, tabserpos[5]) ; }
48   else { ind5 = 0 ; }
49 }
50 listpos();
51 }
52 // =====
53 void servowrite(int sernum, int angl) // pomakni servo motor u zadanu poziciju
54 {
55   int imp = map(angl, MIN_ANG[sernum], MAX_ANG[sernum], MIN_IMP[sernum], MAX_IMP[sernum]); // pretvori kut u impulse
56   pca.writeMicroseconds(sernum,imp);
57   delay(servodelay);
58   tabserpos[sernum] = angl;
59 }
60 // ===== ispiši pozicije servo motora
61- void listpos(){
62
63   String aff = String(tabserpos[0]) + " " + String(tabserpos[1]) + " " + String(tabserpos[2]) + " " + String(tabserpos[3]);
64   String aff1 = String(tabserpos[4]) + " " + String(tabserpos[5]) + " " + String(epractmax[0]) + "-" + String(epractmax[1]);
65   if (lcdpres == 1) {lcdw(2,0,0,""); lcdw(1,0,0,aff); lcdw(1,1,0,aff1); }
66   Serial.println(aff + " " + aff1);
67 }
68 void scenario ()
69 {
70   for (int i=0;i<10;i++) {servomove(prog[0][i],prog[1][i]); }
71 }
72

```

Slika 4.13. Deklaracija servo kontrolera (drugi dio) [22]

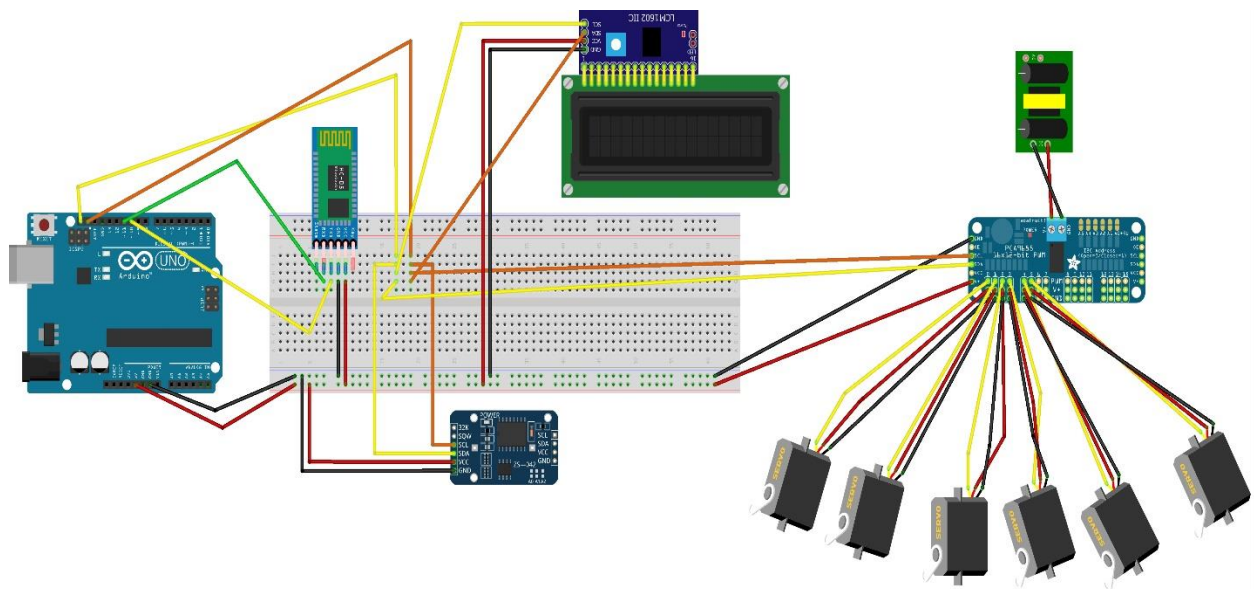
4.6.5. Aplikacija za upravljanje robotskom rukom

Za upravljanje rukom koristimo pametni telefon ili tablet s Android platformom i aplikacijom „Bluetooth Remote“ [22]. Ovu besplatnu aplikaciju je lako konfigurirati. Aplikacija se temelji na izradi programibilnih tipki. Za svaku tipku definiramo znakove za slanje putem Bluetooth-a. Arduino opremljen modul HC05 će primiti informacije i pretvarati ih u akcije. Izmjene panela vrše se ulaskom u način ažuriranja. Svaka se tipka može mijenjati u veličini,

položaju i usmjerenju. Klikom na tipku možemo ažurirati podatke o tipki ili dodati novu tipku. Dva reda tipki s lijeve strane koriste vrijednost „S“ u području „On Press Up“. Kada se gumb otpusti, „S“ se šalje Arduino (koristi se u slučaju „A“ do „L“ programa) [22].

4.7. Postupak spajanja

Na izvor napajanja spajamo modul servo motora (slika 4.14.), minus pol (GND) i plus pol (5 V) zatim koristimo testnu pločicu na kojoj ćemo spojiti ostale komponente sklopa. Servo motore spajamo na priključke servo motora od priključka 0 do priključka 5. Servo modul spajamo na testnu pločicu, potom na Arduino UNO sa SDA i SCL priključcima. AT24 EEPROM memoriju spajamo isto na SDA i SCL priključke na Arduino UNO koristeći testnu pločicu, potom izvor napona na priključak od 5 Volti i minus (GND). Bluetooth modul spajamo sa Arduino UNO-a pomoću testne pločice, priključak 2 na TX i priključak 3 na RX priključak, na ostala dva spajamo izvor napajanja bluetooth modula (5 V i GND). Za kraj spajamo LCD interface na izvor napajanja (5 V i GND) i pomoću testne pločice spojimo priključke SCL i SDA.



Slika 4.14. shema spajanja robotske ruke

5. Mogućnosti primjene projekta

Kad je riječ o aspektu primjene modela robotske ruke, vidimo potencijal u raznim poljima istraživanja i obrazovanja. U obrazovanju možemo primijeniti projekt robotske ruke u svrhu usvajanja znanja i razumijevanja načina programiranja i upravljanja robotima i proizvodnim pogonima u industriji. Projekt robotske ruke ujedinjuje razne znanstvene discipline STEM područja. Praktični dio rada razvija kognitivne vještine matematičkog razmišljanja i računalnog razmišljanja. Odnosno pomažu u razvoju mentalnog procesa koji koristimo za rješavanje problema raznih vrsta kroz uredan slijed radnji. Robotska ruka kojom upravlja Arduino UNO i pokreću servo motori je najbliži praktični primjer industrijskom robotu sa šest osi. Platforma robotske ruke je dobar temelj za eksperimentiranje zbog svoje modularnosti, sa nadogradnjama i raznim modulima. Ovaj projekt može pomoći u razvitku novih načina upravljanja strojevima u industrijskoj proizvodnji. Ako ugradimo PLC kontrolere robotska ruka može raditi i bez Arduino mikrokontrolera. Pomoću Bluetooth veze može u aktualnom vremenom pratiti robota i njegove radnje. Možemo i ugraditi dodatne senzore na samu robotsku ruku koja bi omogućila precizniji rad robota i izmijeniti i dodati nekoliko novih linija koda kako bi povećali njegovu učinkovitost.

6. ZAKLJUČAK

U ovom završnom radu obrađena je tema upravljanja strojevima u industriji. Predstavljena je klasifikacijom i opis raznih vrsta strojeva kojima se u proizvodnji upravlja putem PLC modula i IoT. PLC moduli su bitno uznapredovali od početaka korištenja, sudeći po izvorima informacija, nove vrste PLC modula su u razvoju, pretpostavljamo da će sve više ići prema digitalizaciji modula koji će biti jednostavniji za spajanje i rukovanje a s druge strane moći će upravljati još kompleksnijim sustavima u odnosu na danas, time će se olakšati pristup i servisiranje u slučaju kvarova. IoT će se također sve više razvijati s obzirom na nove tehnologije bežičnog spajanja i prijenosa informacija. Potrebno je i spomenuti razvitak 5G mreže koji će također utjecati na razvoj IoT. Već sada postoje planovi potpunog umrežavanja unutar tvornica, vojske na bojištima, bolnica i slično. Umrežavanje svih sustava značiti će bržu nabavu dijelova i materijala, te značajno povećati učinkovitost proizvodnje. Obradom projektnog zadatka postigli smo nove zaključke. Robotske ruke danas se koriste u mnogim granama industrije, razvitkom pojedinih dijelova robota, pronaći će primjenu i u mnogim drugim granama (primjerice medicini). Arduino razvojna platforma pridonosi ponajviše edukaciji budućih naraštaja inženjera. Pomaže u razvitku razmišljanja i načinu rada s kojim će se susresti na budućem poslu u tvornici. Arduino platforma pruža najbližije iskustvo upravljanju, kontroliranju i programiranju robota u proizvodnji. Smatramo da je potrebno dodatno ulaganje u praktične dijelove nastave, gdje student sudjeluje u nastavi i razvija svoje projekte. Stalni razvoji novih tehnologija predstavlja konstantan izazov za buduće i sadašnje inženjere. Pretpostavljam da će se kurikulum i pristup obrazovanju i pripremi inženjera morati drastično promijeniti i biti konstantno ažurirana, sa naglaskom na praktični dio nastave kao i terenske nastave u kojoj će studenti moći vidjeti buduće poslove iz prve ruke. Također biti će potrebno usmjeriti promoviranje STEM područja mlađim naraštajima, počevši već od mlađih razreda osnovne škole kako bi se učenike zainteresiralo za STEM područja i razvio tehnološki način razmišljanja.

7. LITERATURA

- [1] D. Bajec, Implementacija IoT rješenja u postojeće sustavi upravljanja mikrokontrolerima , Hrvatska, Varaždin, 2020. (17.08.2023.)
- [2] M. Messai Oussama, The industrial and Arduino Bord based Solution for controlling systems, Alžir, 2016. (20.07.2023.)
- [3] D. Novosel, Primjena Raspberry PI u obrazovanju, Hrvatska, Varaždin 2020. (23.07.2023.)
- [4] A. Svedružić, Platforma Arduino iz perspektive učitelja, 2020. Dostupno na: <https://pogledkrozprozor.wordpress.com/2020/12/31/platforma-arduino-iz-perspektive-ucitelja/> (1.08.2023.)
- [5] https://hr.wikipedia.org/wiki/Industrijski_robot (10.08.2023.)
- [6] M. Makovec, Projektiranje simulacije glodanja korištenjem industrijskog robota BB IRB140, Hrvatska, Varaždin, 2019. (15.08.2023.)
- [7] <https://en.wikipedia.org/wiki/Servomotor> (16.08.2023.)
- [9] MME 486, 2006. Introduction to PLCs (18.08.2023.)
- [10] <https://www.ecstuff4u.com/2019/04/advantages-disadvantages-PLC.html> (20.08.2023.)
- [11] <https://projecthub.arduino.cc/danielgass/robot-arm-automation-c4e0cb> (20.08.2023.)
- [12] <https://www.sciencedirect.com/science/article/abs/pii/S1749772821000464> (21.08.2023.)
- [13] D. Mikulić, M. Radujković, D. Palandačić, Potpuno upravljanje kvalitetom u obrazovanju inženjera, Zagreb, Građevinski fakultet, 2011.
- Dostupno na: <https://issuu.com/kvaliteta.net/docs/name882464> (25.08.2023.)
- [14] https://en.wikipedia.org/wiki/Programmable_logic_controller (01.09.2023.)
- [15] <https://www.ecstuff4u.com/2022/11/advantages-of-raspberry-pi.html> (01.09.2023.)
- [16] <https://robu.in/5-pros-and-5-cons-of-raspberry-pi/> (02.09..2023.)
- [17] D. Purković, G. Salopek, Osnove mehatronike: za početno učenje i buduće nastavnike, Rijeka: Odsjek za politehniku Filozofskog fakulteta u Rijeci, 2015. (03.09..2023.)
- [18] <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html> (03.09..2023.)
- [19] <https://www.circuito.io/blog/arduino-code/> (04.09..2023.)
- [20] <https://en.wikipedia.org/wiki/Cobot> (05.09..2023.)
- [21] https://en.wikipedia.org/wiki/Articulated_robot (06.09..2023.)
- [22] <https://projecthub.arduino.cc/danielgass/robot-arm-automation-c4e0cb> (06.09..2023.)

[23] https://en.wikipedia.org/wiki/Robotic_arm (07.09..2023.)

8. POPIS SLIKA

Slika 1.1. Prijenosni strojevi

Slika 1.2. Laki stroj

Slika 1.3. Srednje teški strojevi

Slika 1.4. PLC moduli

Slika 2.1. Raspberry Pi

Slika 3.1. Kartezijanski robot

Slika 3.2. Kolaborativni robot

Slika 3.3. Cilindrični robot

Slika 3.4. Sferični robot

Slika 3.5. SCARA robot

Slika 3.4. Zglobni robot

Slika 3.5. Paralelni robot

Slika 4.1. Bluetooth modul

Slika 4.2. Način spajanja Bluetooth modula

Slika 4.3. EEPROM memorija

Slika 4.4. Servo modul

Slika 4.5. Arduino UNO

Slika 4.6. Dijagram toka

Slika 4.7. glavni dio programa (prvi dio)

Slika 4.8. glavni dio programa (drugi dio)

Slika 4.9. Deklaracija LCD zaslona (prvi dio)

Slika 4.10. Deklaracija LCD zaslona (drugi dio)

Slika 4.11. Deklaracija LCD zaslona (treći dio)

Slika 4.12. Deklaracija servo kontrolera (prvi dio)

Slika 4.13. Deklaracija servo kontrolera (drugi dio)

Slika 4.14. shema spajanja robotske ruke