

# Rekonstrukcija robota kartezijske strukture

---

Major, Krešimir

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:989542>

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

Download date / Datum preuzimanja: **2024-08-24**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)





**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



# **Diplomski rad**

**Krešimir Major**

Zagreb, 2022.



**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



**Diplomski rad**

**Rekonstrukcija robota kartezijske strukture**  
**Reconstruction of the Cartesian robot**

Mentori:

Prof. doc. dr. sc. Tomislav Stipančić

Student:

Krešimir Major

Zagreb, 2022.



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602-14/22-6/1	
Ur. broj: 15-1703-22-	

## DIPLOMSKI ZADATAK

Student: **KREŠIMIR MAJOR**

Mat. br.: 0035211042

Naslov rada na hrvatskom jeziku: **Rekonstrukcija robota kartezijske strukture**

Naslov rada na engleskom jeziku: **Reconstruction of the Cartesian robot**

Opis zadatka:

Razvoj tehnologije diktira uporabu nove opreme koja sa svojim partikularnim svojstvima može doprinijeti postizanju optimalnog rada sustava koristeći koncepte industrije 4.0 na višoj razini uporabe (npr. razine pogona ili tvornice).

Zastarjele robotske osi koje u svojoj kombinaciji čine robota kartezijske strukture moguće je tako rekonstruirati te unaprijediti novim upravljačkim i ostalim komponentama osiguravajući pritom korištenje moderne okoline za programiranje i upravljanje.

U radu je potrebno:

- Odabrati novu upravljačku opremu, detaljno je opisati te usporediti s postojećim (zastarjelim) postavom.
- Osmisliti sheme spajanja te opisati spajanje svih komponenti robota.
- Prikazati te objasniti strukturu PLC programa koristeći CANopen builder programsku okolinu.
- Osmisliti te objasniti na koji način se upravlja te postiže kretanje robota. Objašnjenja potkrijepiti primjerima koji definiraju različita kretanja robota.


Razvijeni robotski postav je potrebno ostvariti koristeći opremu u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava.

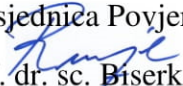
U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:  
29. rujna 2022.

Rok predaje rada:  
1. prosinca 2022.

Predviđeni datum obrane:  
12. prosinca do 16. prosinca 2022.

Zadatak zadao:   
doc. dr. sc. Tomislav Stipančić

Predsjednica Povjerenstva:  
  
prof. dr. sc. Biserka Runje

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se svome mentoru, doc. dr. sc. Tomislav Stipančić, na stručnim savjetima i pruženoj pomoći pri izradi ovoga rada. Isto tako bih se zahvalio asistentu, mag. ing. mech. Leon Koren, na pruženoj pomoći i savjetima pomoću kojih je ovaj rad realiziran i ostvaren.

Zahvalio bih se i svojoj obitelji na velikoj podršci tijekom cjelokupnog studiranja, i ne samo tijekom studiranja nego tijekom svih 18 godina obrazovanja. Hvala vam, bez vas ne bih ostvario ovakve uspjehe.

Krešimir Major

## Sadržaj

<b>Popis slika .....</b>	<b>ii</b>
<b>Popis tablica .....</b>	<b>iii</b>
<b>Popis oznaka .....</b>	<b>iv</b>
<b>Popis mjernih jedinica .....</b>	<b>v</b>
<b>Popis tehničke dokumentacije .....</b>	<b>vi</b>
<b>Sažetak .....</b>	<b>vii</b>
<b>Summary .....</b>	<b>viii</b>
<b>Uvod .....</b>	<b>1</b>
<b>1. Kartezijev koordinatni robot .....</b>	<b>5</b>
<b>2. Postojeća i odabrana upravljačka oprema kartezijevog koordinatnog robota .....</b>	<b>8</b>
2.1 Postojeća oprema .....	8
2.2 Odabrana oprema .....	11
2.3 Usporedba postojeće i odabrane opreme .....	15
<b>3. Sklapanje svih komponenti robota i elektroničke sheme spajanja .....</b>	<b>16</b>
3.1 Komponente robota .....	16
3.2 Elektroničke sheme spajanja .....	20
Programabilni logički kontroler – PLC .....	22
Servo pogoni .....	22
Motori .....	23
<b>4. PLC program za upravljanje robota .....</b>	<b>24</b>
4.1 Programski kod PLC-a (Ladder diagram) .....	24
<b>5. Upravljanje robotom i primjer G-koda .....</b>	<b>31</b>
<b>6. Zaključak .....</b>	<b>36</b>
<b>Literatura .....</b>	<b>37</b>
<b>Prilozi .....</b>	<b>38</b>

## Popis slika

Slika 1 – 6-osni zglobovi robot (eng. Articulated robot).....	2
Slika 2 – Podjela robota s obzirom na strukturu.....	3
Slika 3 – Usporedba radnog prostora kartezijskog (lijevo) i SCARA) (desno) robota.....	3
Slika 1.1 - Kartezijski koordinatni robot .....	5
Slika 1.2 – Primjer položaja osi (linearnih zglobova) kartezijskog koordinatnog robota .....	6
Slika 1.3 – Primjeri kartezijskih koordinatnih robota .....	7
Slika 2.1.1 – Kontroler pokreta IAI X-SEL Controller .....	9
Slika 2.1.2 – Aktuator serije ISA/ISPA .....	9
Slika 2.1.3 – CAD prikaz aktuatora x-osi.....	10
Slika 2.2.1 – Programabilni logički kontroler DVP15MC11T .....	11
Slika 2.2.2 - Servo pogoni snaga 200 W (lijevo) i 100 W (desno).....	13
Slika 2.2.3 - Servo motori snaga 100 W (lijevo) i 200 W (desno).....	14
Slika 3.1.2 – Prikaz cijelog kartezijskog koordinatnog robota .....	17
Slika 3.1.3 – Sklop komponenti na aktuatoru X-osi .....	18
Slika 3.1.4 – Distancer Z-osi (lijevo) i distancer X i Y-osi (desno).....	18
Slika 3.1.5 – Sklop komponenti na R-osi .....	19
Slika 3.1.6 – Sklop spojke i harmonijskog prijenosnika.....	19
Slika 3.1.7 – CAD prikaz kućišta motora (lijevo) i držača krajnjeg prekidača (desno) za X-os. 20	20
Slika 3.2.1 – Položaj elektro ormara .....	21
Slika 3.2.2 – Elektro ormar .....	21
Slika 3.2.3 – Prikaz priključaka na PLC-u .....	22
Slika 3.2.4 – Prikaz priključaka na servo pogonu .....	23
Slika 4.1.2 – Programski kod, Linija 1.....	25
Slika 4.1.3 – Programski kod, Linije 4 i 5 .....	26
Slika 4.1.4 – Programski kod, Linija 6.....	26
Slika 4.1.5 – Programski kod, Linije 11, 12, 13 .....	27
Slika 4.1.6 – Programski kod, Linija 18.....	27
Slika 4.1.7 – Programski kod, Linije 20 i 21 .....	28
Slika 4.1.9 – Programski kod, Linija 19.....	29
Slika 4.1.10 – Programski kod, Linija 23.....	30
Slika 5.1 – Početni prozor grafičkog sučelja.....	32
Slika 5.2 – Prozor grafičkog sučelja nakon povezivanja s PLC-om .....	32
Slika 5.3 – Prozor grafičkog sučelja nakon paljenja osi.....	33
Slika 5.4 – Prozor grafičkog sučelja tijekom upravljanja robotom .....	34
Slika 5.5 – Prozor grafičkog sučelja pri upisu varijabli .....	34
Slika 5.6 – Primjer G-koda .....	35

## Popis tablica

Tablica 2.3.1 - Usporedba trenutne i nove opreme.....	15
Tablica 3.1.1 – Popis opreme kartezijevog koordinatnog robota.....	16
Tablica 4.1.1 – Značajke osi X, Y, Z i R .....	25
Tablica 4.1.8 – Popis varijabli za definiranje kinematike gibanja robota .....	29



## **Popis oznaka**

PLC – programabilni logički kontroler

SCARA - Zglobna robotska ruka selektivne usklađenosti (eng. Selective Compliance Articulated Robot Arm)

IAI - International Automation Industry

CAN – controller area network

ISO – International Organization for Standardization

## Popis mjernih jedinica

Oznaka	Mjerna jedinica	Parametar
<i>m</i>	<i>kg</i>	<i>masa</i>
<i>d</i>	<i>m</i>	<i>duljina</i>
	<i>B</i>	<i>memorija</i>
<i>t</i>	<i>s</i>	<i>vrijeme</i>
<i>P</i>	<i>W</i>	<i>snaga</i>
<i>f</i>	<i>Hz</i>	<i>frekvencija</i>
<i>n</i>	<i>min<sup>-1</sup></i>	<i>broj okretaja</i>
<i>v</i>	<i>mm/s</i>	<i>brzina</i>
<i>a</i>	<i>mm/s<sup>2</sup></i>	<i>akceleracija</i>
<i>U</i>	<i>V</i>	<i>napon</i>

## Popis tehničke dokumentacije

- Poz – 1      Držac krajnjeg prekidača
- Poz – 2      Distancer X-osi
- Poz – 3      Kučište X-osi

## Sažetak

U ovom radu bavimo se rekonstrukcijom kartezijevog koordinatnog robota. Potreba za samom rekonstrukcijom došla je zbog zastarjelosti opreme i korisničkog sučelja koji su na robotu bili implementirani nekoliko desetljeća i koji se danas više nigdje u industriji ne koriste. Kroz rad obuhvatit će se tri velike cjeline kroz koje je izvršena rekonstrukcija samog robota:

- I. Odabir nove opreme
- II. Implementiranje nove opreme na robota
- III. Programiranje robota i osposobljavanje njega za daljnji razvoj

Kroz prvu cjelinu prikazat će se i opisati nova, ali i zastarjela oprema robota. Detaljno će se opisati specifikacije svake komponente, a na kraju će se usporediti sve komponente zastarjele opreme, koje se zamjenjuju, s novim komponentama.

U drugoj cjelini opisat će se implementacija novih komponenti na postojeći robot. Prikazat će se njihovi položaji i objasniti razlog i način implementacije. Osim samog postavljanja komponenti potrebno je i elektronički ih spojiti. Sheme elektroničkog spajanja možete vidjeti u prilogima na kraju rada, a opis spajanja možete pročitati u poglavlju 3.

Zadnji dio rekonstrukcije kartezijevog koordinatnog robota obuhvaća programiranje samog robota. U poglavljima 4 i 5 možete vidjeti detaljan opis i prikaz programa potrebnih kako bi se robot osposobio za rad, ali i za daljnji razvoj.

Robot se nalazi u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava, na Fakultetu strojarstva i brodogradnje, i nakon završetka rekonstrukcije koristi se za daljnje edukacijske svrhe.

Ključne riječi: kartezijev, koordinatni, robot, aktuator, servo pogon, motor, program, programski kod, elektronička shema, osi, linearni, programabilni logički kontroler, Delta, CANopen, ethernet, CANopen Builder

## Summary

In this paper, we deal with the reconstruction of the Cartesian coordinate robot. The need for the reconstruction itself came about due to the obsolescence of the equipment and the user interface that had been implemented on the robot for several decades and are no longer used anywhere in the industry today. The work will cover three large chapters through which the reconstruction of the robot itself was carried out:

- I. Selection of new equipment
- II. Implementation of new equipment on the robot
- III. Programming the robot and preparing it for further development

The first part will show and describe new and also outdated robot equipment. The specifications of each component will be described in detail, and at the end, all the components of the obsolete equipment, which are being replaced, will be compared with the new components.

The second part will describe the implementation of new components on the existing robot. Their positions will be shown and the reason and method of implementation will be explained. In addition to installing the components, it is also necessary to connect them electronically. You can see the electronic connection schemes in the appendices at the end of the paper, and you can read the description of the connection in chapter 3.

The last part of the reconstruction of the Cartesian coordinate robot includes the programming of the robot itself. In chapters 4 and 5, you can see a detailed description and presentation of the programs needed to prepare the robot for work, as well as for further development.

The robot is located in the Laboratory for the design of manufacturing and assembly systems and after the completion of the reconstruction, it is used for further educational purposes.

Keywords: Cartesian, coordinate, robot, actuator, servo drive, motor, program, program code, electronic scheme, axes, linear, programmable logic controller, Delta, CANopen, ethernet, CANopen Builder

## Uvod

Roboti su automatizirani strojevi višestruke namjene koji se sastoje od konstrukcije s pripadajućim pogonskim uređajima, senzora i upravljačkog uređaja. Dijelev se po stupnju pokretljivosti (statički i mobilni roboti), strukturi konstrukcije (mehatronički, biotronički i bioroboti), namjeni (industrijski, medicinski, edukacijski, podvodni, roboti za istraživanje svemira, vojni roboti, osobni roboti), veličini (makroroboti, mikroroboti i nanoroboti).

Robotika je interdisciplinarno znanstveno područje zasnovano na mehatronici, a bavi projektiranjem, konstruiranjem, upravljanjem i primjenom robota. Robotikom objedinjena su znanstvena područja i discipline poput strojarstva, elektrotehnike, elektronike, automatike, računarstva i umjetne inteligencije.

Prema ISO 8373 robot je automatski upravljani, programabilni, višenamjenski manipulator koji se može programirati u tri ili više osi, koji može biti stacionaran ili mobilan. S humanitarnog gledišta, industrijski roboti su idealni za poslove koji se smatraju teškim i neprikladnim za ljude, te za poslove koji se ponavljaju više puta i kao takvi se smatraju monotonim. Također, roboti se koriste i u onim procesima gdje se traži visoka i ujednačena kvaliteta te velika produktivnost.

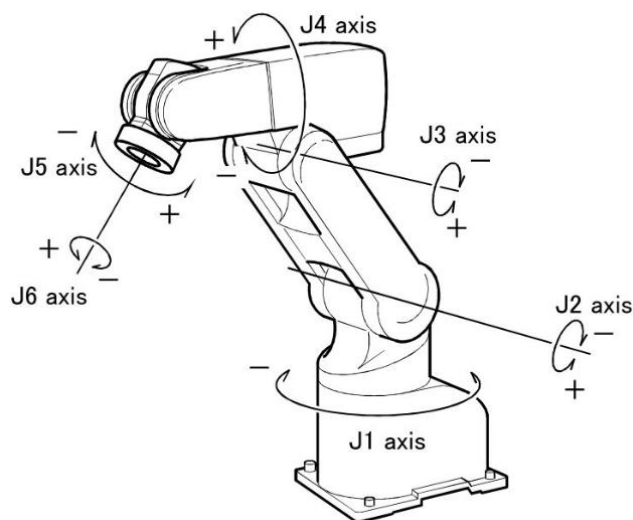
Industrijski robot je robotski sustav koji se koristi za proizvodnju. Industrijski roboti su automatizirani, programabilni i sposobni za kretanje po tri ili više osi. Tipične primjene robota uključuju zavarivanje, bojanje, sastavljanje, rastavljanje, odabiranje i postavljanje tiskanih ploča, pakiranje i označavanje, paletiziranje, pregled proizvoda i testiranje. Sve se postiže velikom izdržljivošću, brzinom i preciznošću. Oni mogu pomoći u rukovanju materijalom.

Najčešće industrijske strukture robota su kartezijska, cilindrična, kvazi cilindrična, sferna, rotacijska, SCARA i heksapodna. Ono što razlikuje svaku strukturu od drugih je vrsta i položaj svake osi na robotu. Postoje dvije vrste osi: translacijska (linearna) T i rotacijska R, i upravo kombinacijom takvih osi dobijemo različite strukture robota.

- Kartezijska struktura – 3 translacijske (linearne) osi (TTT)
- Cilindrična struktura – 2 translacijske i 1 rotacijska os (RRT)
- Kvazi cilindrična struktura – 1 translacijska i 2 rotacijske osi (RTR)
- Sferna struktura - 1 translacijska i 2 rotacijske osi (RRT)
- Rotacijska struktura – 3 rotacijske osi (RRR)
- SCARA struktura – 1 translacijska i 3 rotacijske osi (RRRT)
- Zglobna struktura – 6 rotacijskih osi

Kartezijski robot koji se naziva i pravocrtni, portalni robot (eng. Gantry robot) i x-y-z robot je detaljnije prikazan i opisan u sljedećim poglavljima jer se u ovome radu bavimo rekonstrukcijom kartezijskog koordinatnog robota.

Zglobni roboti (eng. Articulated robot) su najčešći industrijski roboti. Izgledaju poput ljudske ruke, zbog čega se nazivaju i robotska ruka ili ruka manipulatora. Njihova konstrukcija s nekoliko stupnjeva slobode omogućuje zglobnim rukama širok raspon pokreta. Ovakav robot koristi svih šest okretnih zglobova za pristup svom radnom prostoru. Obično su zglobovi raspoređeni u "lanac", tako da jedan zglob podupire drugi dalje u lancu.

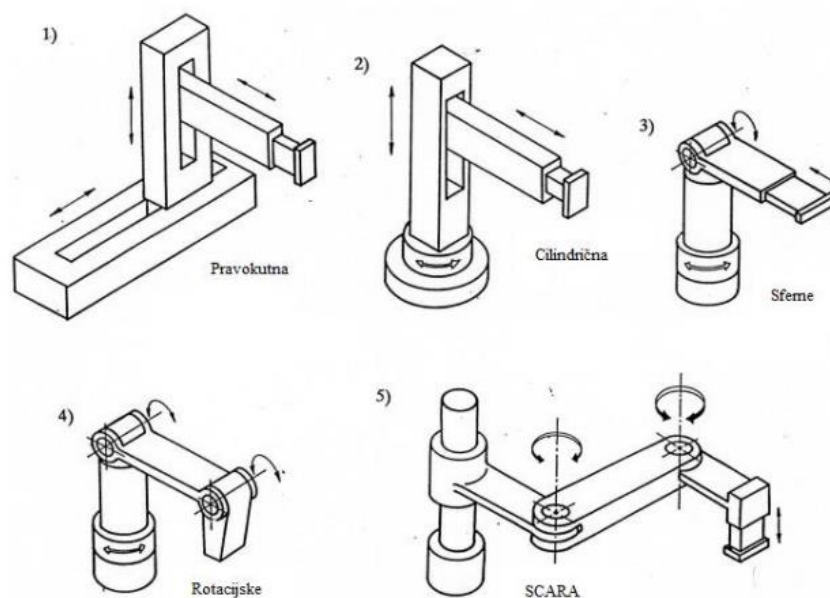


**Slika 1 – 6-osni zglobni robot (eng. Articulated robot)**

Cilindrične robote karakterizira njihov rotacijski zglob u podnožju i dva linearna zgloba. Klizanjem linearni zglobovi se mogu kretati okomito i vodoravno. Kompaktni dizajn omogućuje robotu da dosegne uske radne prostore bez gubitka brzine.

Sferni roboti imaju 1 translacijski i dva rotacijska zgloba. Oni su jedni od prvih robota koji su se koristili u industrijskim aplikacijama. Oni su stacionarne robotske ruke sa sfernim ili gotovo sferičnim radnim omotačima koje se mogu postaviti u polarnom koordinatnom sustavu. Ponekad se smatraju polarnim robotima. Obično se koriste za strojno održavanje kod tlačnog lijevanja, brizganja plastike i ekstruzije te za zavarivanje.

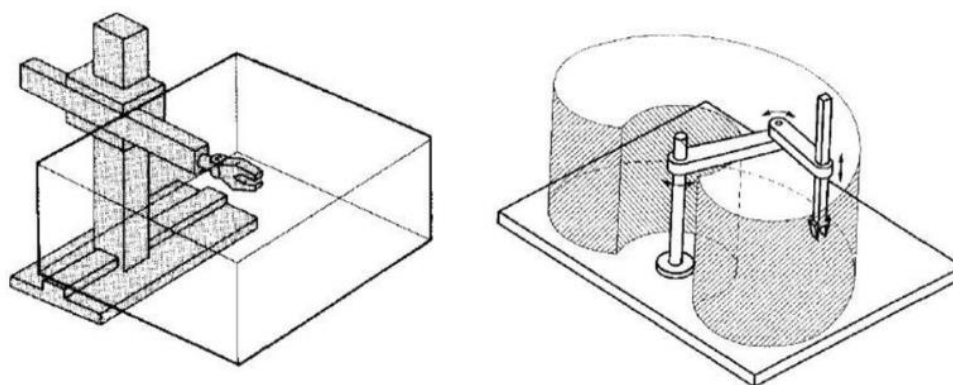
SCARA je akronim za eng. Selective Compliance Assembly Robot Arm. SCARA roboti se prepoznaju po dva paralelna zgloba koji omogućuju kretanje u ravnini X-Y. Rotirajuće osovine postavljene su okomito na bazu robota. Drugi atribut SCARA robota je zglobna dvostruka ruka koja je slična ljudskoj ruci, otuda se i često nazivaju zglobni. Ova značajka omogućuje ruci da se ispruži u ograničena područja, a zatim uvuče ili preklopi u stranu. Ovo je korisno za prijenos dijelova iz jedne ćelije u drugu ili za utovar ili istovar zatvorenih procesnih stanica. SCARA roboti se koriste za poslove koji zahtijevaju precizne bočne pokrete. Idealni su za montažne primjene.



**Slika 2 – Podjela robota s obzirom na strukturu**

Osim položaja osi, strukture se međusobno razlikuju i po prostoru kojeg roboti zauzimaju. Prostor kojeg zauzimaju se dijeli u više skupina: ukupni prostor, prostor krajnjeg dohvata i radni prostor.

Ukupni prostor je prostor kojeg robot zauzima zajedno sa cijelom svojom konstrukcijom i postoljem (ako ga ima) na kojem se nalazi. Prostor krajnjeg dohvata je volumen u prostoru u kojem se robot sa svojim vrhom može nalaziti. Cijeli taj volumen nije i radni prostor. Primjer razlike između ta dva prostora je rad robotske ruke, koja se može u potpunosti ispružiti vodoravno sa svojim postoljem i doći u najudaljeniju točku, no u tom položaju robot nema nikakvu nosivost i samim time taj položaj je nepoželjan i ne koristi se tijekom rada robota. Radni prostor robota je volumen u prostoru u kojem robot izvršava svoje naredbe.



**Slika 3 – Usporedba radnog prostora kartezijskog (lijevo) i SCARA (desno)**



U ovome radu bavimo se robotom kartezijske strukture. Riječ je o robotu koji se nalazi u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava i koji se koristi već nekoliko desetljeća. Njegova izvorna primjena je bila da se nalazi iznad proizvodne trake po kojoj putuju određeni proizvodi te onda pomoću hvataljke prima i pomiče dijelove u određeni položaj. Tijekom rada prikazat će se i detaljno opisati izgled samog robota te će se opisati i njegova funkcionalnost. Oprema kartezijskog koordinatnog robota je oprema tvrtke IAI. Tvrtka IAI osnovana je u Japanu 1976., s američkim sjedištem osnovanim 1989. IAI dizajnira, proizvodi i prodaje kompletnu liniju sustava za kontrolu kretanja. Kao istaknuti svjetski lider u proizvodnji linearnih električnih aktuatora i SCARA robota niske cijene i visokih performansi, osiguravaju više od polovice svih linearnih aktuatora prodanih u Japanu i vodeći su izvor napredne uključi i pokreni (eng. plug & play) automatizacije.

Zbog zastarjelosti opreme kartezijskog robota došlo je do potrebe za rekonstrukcijom samog robota. Programabilni logički kontroler IAI X-SEL Controller na tržištu se je pojavio 90-tih godina prošlog stoljeća. Osim kontrolera, za rad kartezijskog robota u laboratoriju, koristili su se i servo motori tvrtke IAI, linearni aktuatori i njihovo korisničko sučelje potrebno za programiranje kontrolera i upravljanje robotom.

Kartezijski koordinatni robot u laboratoriju fakulteta koristi se za edukaciju, ali kako bi bio podoban za daljnje korištenje potrebno je napraviti kompletnu rekonstrukciju robota kako bi ga se prilagodilo današnjim uvjetima. Potrebno je zamijeniti sve „zastarjele“ komponente i zamijeniti ih „modernijima“. Proces kompletne rekonstrukcije sastoji se od nekoliko faza:

1. Faza: Odabir nove opreme
2. Faza: Implementiranje nove opreme na robota
3. Faza: Programiranje robota i osposobljavanje njega za daljnji razvoj

Odabrana nova oprema, za izvršenje rekonstrukcije robota, je oprema tvrtke Delta electronics. Delta Electronics (poznata i kao DELTA) je tajvanska tvrtka za proizvodnju elektronike. Detaljni prikaz odabrane nove opreme slijedi u idućim poglavljima u kojima će te biti upoznati s tehničkim specifikacijama svakog elementa, ali i detaljno će se prikazati njihova hardverska i softverska implementacija na robota

Ovim putem zahvalio bih se firmi Proton EL d.o.o. zbog njihove donacije opreme, firme Delta electronics, laboratoriju za projektiranje izradbenih i montažnih sustava, čime su nam omogućili da izvršimo cjelovitu rekonstrukciju kartezijskog koordinatnog robota.

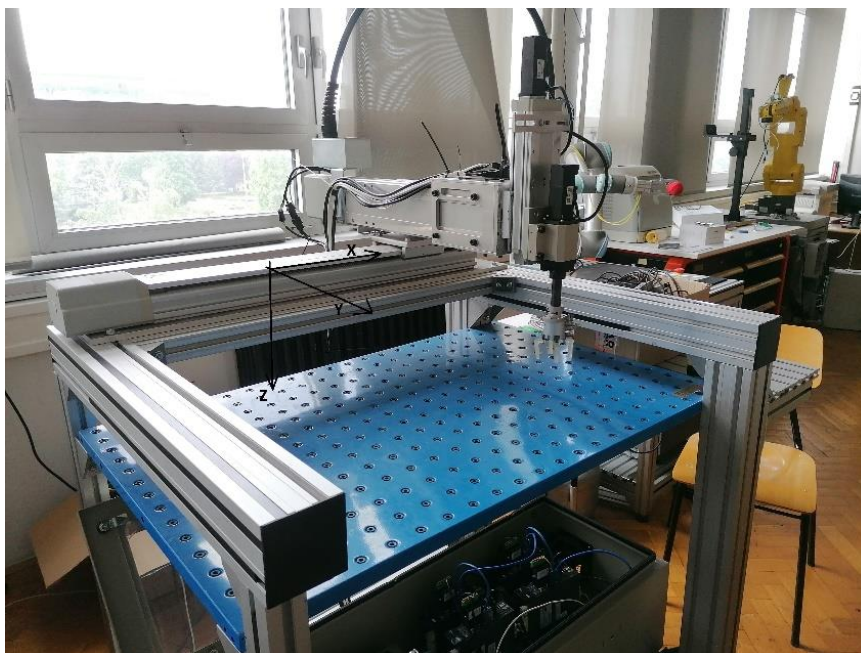
Cijeli postupak rekonstrukcije robota, zamjena zastarjelih komponenti s novima i integracija novih komponenti s ostatkom robota, detaljno je prikazan kroz poglavlja u ovom radu. U 2. i 3. poglavlju možete vidjeti izbor i opis novih i starih komponenti te možete vidjeti detaljan prikaz i način spajanja svih komponenti kartezijskog robota. Kasnije u poglavljima 4 i 5 možete vidjeti način programiranja samog robota kako bismo postigli mogućnost automatiziranog upravljanja robotom. Za kraj, svi potrebni podaci za cjelovitu rekonstrukciju robota možete vidjeti u poglavljima kroz rad i na kraju samog rada u prilogima.

# 1. Kartezijev koordinatni robot

Roboti imaju mehanizme koji se sastoje od krutih karika međusobno povezanih zglobovima s linearnim ili rotacijskim (revolucijskim) kretanjem ili kombinacijom ta dva. Aktivni linearni i aktivni revolucijski zglobovi pokreću se motorima pod programabilnom kontrolom za upravljanje objektima radi obavljanja složenih automatiziranih zadataka. Linearno gibanje aktivnih linearnih spojeva može se pokretati rotacijskim motorima preko zupčanika ili remenica.

Kartezijev koordinatni robot (poznat i kao linearni robot) je industrijski robot čije su tri glavne osi upravljanja linearne (tj. kreću se pravocrtno umjesto da se okreću) i nalaze se pod pravim kutem u međusobnim odnosima. Pokreti ostvareni trima linearnim zglobovima, koji omogućuju tri vrste pomaka (lijevo-desno, naprijed-nazad i gore-dolje), ostvaruju se u kartezijevom koordinatnom sustavu. Linearni zglobovi su poravnati s X, Y i Z osi kartezijevog koordinatnog sustava. Ovakav mehanički raspored osi pojednostavljuje korištenje upravljačke ruke robota. Ovakvi roboti imaju visoku pouzdanost i preciznost pri radu u trodimenzionalnom prostoru i iako nisu striktno „roboti“, druge vrste manipulatora, kao što su računalno numerički upravljani (CNC) strojevi, 3D pisači ili crtači s perom, također imaju isti mehanički raspored međusobno okomitih aktivnih linearnih spojeva.

Osim triju glavnih osi moguće je uključiti i rotacijsku os (R) koja se ponekad uključuje kao pomoć za robotsku ruku i nalazi se na najudaljenijoj osi kartezijevog robota. U slučaju robota o kojem pričamo u ovom radu, on sadrži rotacijsku os koja se nalazi na Z osi i služi za dodatno upravljanje hvataljkom robota. Zbog toga radi se o 4-osnom kartezijevom robotu (X, Y, Z i R). Prikaz robota možete vidjeti na slici 1.1.



Slika 1.1 - Kartezijev koordinatni robot

Ono što kartezijskog koordinatnog robota čini robotom je mogućnost koordiniranog i sinkroniziranog upravljanja svim osima robota. Kod nekih robota, kao kod robota sa slike 1.1, moguće je odvojiti rotacijsku os R od ostalih osi, kako bi se ostvarila mogućnost rotacije osi neovisno o trenutnoj kretnji ostalih osi. Mogućnost sinkronizacije i upravljanje osima ostvarena je kroz zajednički kontroler pokreta (eng. Motion controller). Detaljniji opis postojeće i buduće upravljačke opreme slijedi u idućim poglavljima.

Kartezijski roboti imaju nekoliko prednosti u odnosu na SCARA i 6-osne tipove. Prvo, kartezijski dizajn pruža pravokutni radni prostor u kojem se značajan postotak koristi kao aktivno radno područje robota. SCARA i 6-osni tipovi robota, s druge strane, imaju kružne ili ovalne radne prostore koji često rezultiraju s puno mrtvog (neiskorištenog) prostora, posebno kada je potreban hod ili doseg vrlo velik. Glavni nedostatak kartezijskih robota je što im je potreban veliki volumen prostora za rad, iako se cijeli prostor ne koristi. Kartezijski robot zauzima isto toliko prostora koliko i X-os. Ako, na primjer, imate radni prostor koji zahtijeva 600 mm hoda, X-os će zauzeti taj prostor. Kartezijski roboti zauzimaju najveću potrebnu površinu od svih konfiguracija robota. Izložene površine za vođenje zahtijevaju pokrivanje u korozivnim ili prašnjavim okruženjima i ne mogu raditi pod vodom, kao što to mogu činiti SCARA roboti.



**Slika 1.2 – Primjer položaja osi (linearnih zglobova) kartezijskog koordinatnog robota**

Osi kartezijskog robota izrađene su iz nekog oblika linearnog pokretača, bilo kupljenog kao unaprijed sastavljeni sustav od proizvođača ili po narudžbi, sastavljenog od linearnih vodilica i pogonskih komponenti. Kartezijski roboti mogu se konstruirati i izraditi iz gotovo bilo koje vrste linearnog pogona s bilo kojim različitim mehanizmima pogona – remenom, kugličnim vijkom, pneumatskim aktuatorom ili linearnim motorom (imajte na

umu da su također mogući pogoni sa zupčastom letvom i zupčanikom, ali se češće koriste u portalnim sustavima s vrlo dugim hodovima). Zato oni često imaju bolju točnost pozicioniranja i ponovljivosti od SCARA i 6 – osnih (zglobnih) robota.

Kartezijevi roboti također imaju prednost u jednostavnijem korištenju u pogledu programiranja jer je njihova kinematika jednostavnija (ima samo 3 kartezijeve osi, bez rotacijskih). Zbog svoje krute strukture ovi roboti mogu manipulirati s relativno velikim teretima pa se često koriste za aplikacije podizanja i postavljanja, utovar alatnih strojeva i slaganje dijelova u spremnike.

Popularne primjene za robote s kartezijevim koordinatama su računalni numerički upravljački stroj (CNC stroj) i 3D ispis. Najjednostavnija primjena koristi se u strojevima za glodanje i crtačima gdje se alat, kao što je olovka, premješta preko X-Y ravnine i podiže i spušta na površinu kako bi se stvorio precizan dizajn. Eng. „Pick and place“ strojevi još su jedna primjena za robote s kartezijevim koordinatama. Na primjer, kartezijevi roboti iznad glave primjenjuju se za kontinuirani utovar i istovar dijelova na proizvodnim linijama CNC tokarilica, izvodeći operacije skupljanja i postavljanja teških tereta u 3 osi (X, Y, Z) s velikom brzinom i velikom preciznošću pozicioniranja. Općenito, kartezijevi roboti iznad glave prikladni su za mnoge sustave automatizacije.



Slika 1.3 – Primjeri kartezijevih koordinatnih robota

## 2. Postojeća i odabrana upravljačka oprema kartezijskog koordinatnog robota

Kao i kod svakog robota, tako i kod našeg kartezijskog koordinatnog robota, moramo imati opremu potrebnu za upravljanje robotom. Glavne upravljačke komponente korištene kod kartezijskog robota su programabilni logički kontroler, servo pogoni (eng. Servo drives) i motori. Osim glavnih komponenti, bez kojih rad robota ne bi bio moguć, na robotu se koriste dodatna oprema, u pravilu senzori, koji omogućuju robotu, ali i korisniku, precizniji, bolji i brži rad. Detaljni opis i prikaz trenutne i nove opreme slijedi u nastavku ovog poglavlja.

### 2.1 Postojeća oprema

Trenutni kontroler koji se koristi za upravljanje kartezijskim robotom je *IAI X-SEL Controller*. IAI (eng. International Automation Industry), kompanija koja je proizvela ovaj kontroler, je najveći proizvođač kartezijskih koordinatnih robota. Upravljanje robotom preko IAI X-SEL kontrolera vrši se putem programskih operacija. Programski jezik koji X-SEL kontroleri koriste je jednostavan i mogu se kompleksne radnje robota isprogramirati. Kontroleri omogućavaju komunikaciju preko mreža DeviceNet, CC-Link, ProfiBus i Ethernet. Imaju veliku brzinu procesiranja podataka i zbog toga ostvaruju veliku preciznost u radu robota pri većim brzinama, uz to omogućuju sinkronizaciju dvaju aktuatora na osima robota s čime se ostvaruje sinkronizirani rad dviju osi robota.

IAI X-SEL-K kontroler, vrsta kontrolera koji se koristi za upravljanje kartezijskim robotom sa slike 1.1, za napon napajanja može koristiti jednofaznih 100 ili 200 V. Koristi se za upravljanje s 1 do 4 osi, a snage motora za upravljanje osima mogu varirati od 20 do 750 W. Aktuatori koji se koriste za kretanje osi, koji će kasnije biti prikazani, dolaze iz ISA/ISPA serija. Kontroler za određivanje pozicije robota može koristiti inkrementalnu ili apsolutnu metodu. Broj programa koji mogu istovremeno biti na kontroleru iznosi 64, a broj koraka koje pojedini program može sadržavati iznosi 6 000. Broj pozicija robota koje možemo ostvariti kroz programe iznosi 3 000, a samo sučelje u kojem se pišu programi može biti preko kontrolne kutije (eng. Teach Pendant) ili preko računalnog softvera IA-101-X-MW. Kontroler koji se trenutno koristi na kartezijskom robotu prikazan je na slici 2.1.1.

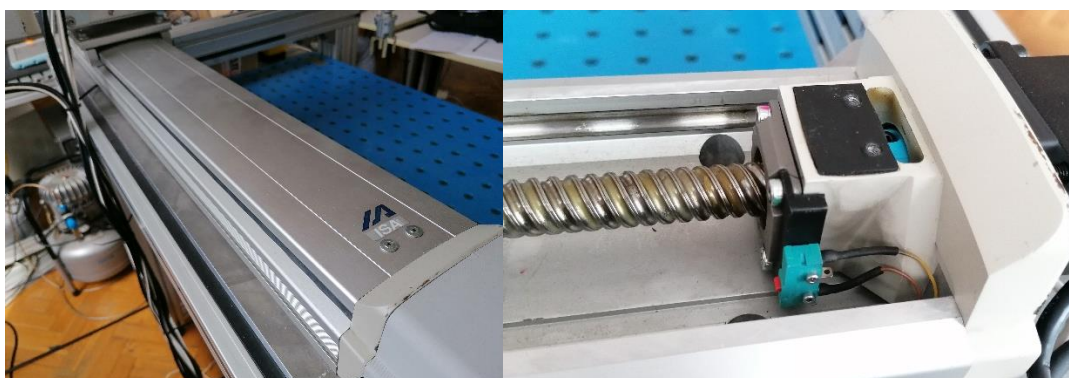




**Slika 2.1.1 – Kontroler pokreta IAI X-SEL Controller**

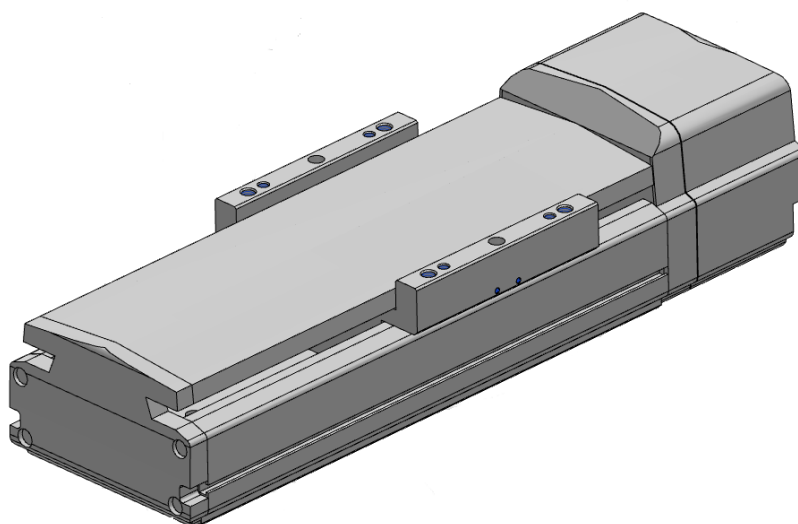
Trenutni kontroler, prikazan na slici 2.1.1, punog naziva i oznake glasi: **IAI X-SEL-K-4-150IB-60I-60IB-30RIL-N1-EEE-5-1**. Iz oznake možemo iščitati kako se radi o kontroleru za opću upotrebu (**K**) koji služi za upravljanje s 4 osi (**4**). Na prvoj osi (**X**) se nalazi servo motor snage 150W, pozicija osi određuje se inkrementalnim enkoderom i sadrži svoju kočnicu (**150IB**). Druga i treća os (**Y** i **Z**) imaju motor snage 60W, pozicije im se određuju inkrementalnim enkoderom i **Z** os još sadrži svoju kočnicu, (**60I-60IB**). Četvrta, rotacijska os, (**R**) ima motor snage 30W, također se inkrementalnim enkoderom određuje pozicija, i ona je došla s ugrađenim senzorom za početni položaj (**30RIL**). Kontroler pokreta sadrži 32 ulaza i 16 izlaza (**N1**), dolazi s 5 m kabela za spajanje s računalom (**5**) i njegov napon napajanja iznosi 100V (**1**).

Osim kontrolera važan dio opreme su i aktuatori preko kojih kontroleri ostvaruju željene kretnje robota. Kartezijevi koordinatni roboti, zbog toga što u sve tri osi imaju translacijsko gibanje, koriste linearne aktuatore. Linearni aktuator obično se nalazi između dva linearna ležaja razmaknuta jedan od drugoga da podrže momentna opterećenja. Primjer takvog aktuatora korištenog na našem kartezijevom robotu možete vidjeti na slici 2.1.2.



**Slika 2.1.2 – Aktuator serije ISA/ISPA**

Aktuatori serije ISA/ISPA (slika 2.1.2), koji se koriste za pokretanje osi, sastoje se od baze, linearnih vodilica, kugličnog vijka i AC servo motora. Ovi aktuatori imaju veći maksimum akceleracije i deceleracije, u odnosu na prijašnje serije aktuatora, u iznosu od 1 G ( $9800 \frac{mm}{s^2}$ ). Svaka os ima svoj aktuator i koji se međusobno razlikuju. Za X-os koristi se aktuator oznake ISA-MXM. Taj aktuator koristi servo motor snage 150 W, položaj osi se određuje inkrementalnim enkoderom i za svaki puni okret kugličnog vijka klizač se pomakne za 10 mm. Ukupni pomak po X-osi koji se može ostvariti iznosi 585 mm i maksimalna brzina kretanja klizača koja se može ostvariti iznosi  $500 \frac{mm}{s}$ . Maksimalna opterećenja koja os može izdržati iznose u horizontalnom smjeru 80 kg, a u vertikalnom 19 kg.



**Slika 2.1.3 – CAD prikaz aktuatora x-osi**

Za Y-os koristi se aktuator oznake ISA-SYM koji koristi servo motor snage 60 W. Položaj se također određuje inkrementalnim enkoderom i za svaki puni okret kugličnog vijka klizač se pomakne za 4 mm. Maksimalna brzina klizača koja se može ostvariti iznosi  $200 \frac{mm}{s}$  na radnom području od 385 mm. Opterećenja koja os može izdržati iznose 50 kg u horizontalnom smjeru i 14 kg u vertikalnom smjeru. Za treću, Z-os, koristi se aktuator tipa ISA-SZM, servo motor snage 60 W i također se pozicija osi određuje inkrementalnim enkoderom. Za svaki puni okret kugličnog vijka klizač se pomakne za 4 mm, a maksimalna brzina kretanja klizača iznosi  $200 \frac{mm}{s}$ . Ovakva vrsta aktuatora, za Z-os, može se isključivo koristiti kao vertikalna os. Aktuator dolazi s već ugrađenom kočnicom i maksimalno opterećenje koje os može izdržati iznosi 14 kg, a ukupni pomak koji se može ostvariti iznosi 90 mm. Ukupna duljina svakog aktuatora ovisi o ukupnom pomaku koji se može ostvariti na toj osi, dok širina aktuatora ovisi o kojoj se standardnoj veličini radi, pa tako aktuator X-osi koji je srednje veličine ima širinu iznosa 120 mm dok aktuatori Y i Z-osi imaju širinu 90 mm.

Prilikom spajanja aktuatora s IAI X-SEL kontrolerom za svaki je potrebno spojiti 3 kabela: kabel napajanja motora, kabel enkodera i kabel prekidača za krajnji položaj.

## 2.2 Odabrana oprema

Zbog zastarjelosti opreme opisane i prikazane u poglavlju 2.1 Postojeća (zastarjela) oprema, donesena je odluka o zamjeni s novijom i aktualnijom. Oprema kompanije IAI (eng. International Automation Industry ) koristi se već preko 20 godina i njezino korisničko sučelje je zastarjelo i ima sve manju primjenu u svijetu, zbog toga odluka je pala na postavljanju novije opreme na kartezijev robot. Riječ je o opremi kompanije Delta Electronics, konkretnije, upotrebljeni su njihov programabilni logički kontroler, servo pogoni (eng. Servo drives) i motori.

Deltin programabilni logički kontroler (PLC) dolazi iz njihove serije kontrolera DVP-MC. Iz te serije dolaze PLC-ovi za višeosno kontroliranje pokreta konstruirani za mrežnu komunikaciju preko komunikacijskih priključaka CANopen i Ethernet. DVP-MC podržava komunikaciju preko CANopen i Ethernet s već ugrađenim instrukcijama za kontrolu pokreta, BufferMode i Jerk. Ovi PLC-ovi mogu upravljati do 24 realnih osi preko priključka za kretanje (eng. Motion port), također podržavaju instrukcije za jednoosno upravljanje pokreta kao što su brzina, pozicija, moment (eng. Torque), vraćanje u početni položaj (eng. Homing) i instrukcije za višeosno upravljanje kao što su elektronički zupčanik, elektronička kamera (E-Cam), rotacijski rez i G-kod.

DVP-MC serija PLC-a ima više ugrađenih komunikacijskih sučelja i može se jednostavno povezati s drugom opremom bez dodatnih komunikacijskih modula. Osim komunikacijskih priključaka CANopen i Ethernet, uz dodatne module za proširenje PLC se može povezivati s raznim uređajima za industrijsku automatizaciju putem EtherCAT, DeviceNet, ProfiBus i RS-485 priključaka. Upotreba ovih kontrolera može imati široku primjenu u raznim industrijama upravljanja automatizacijom kao što su pakiranje, tisak, kapsuliranje, rezanje žice, proizvodnja lijekova i tako dalje.



Slika 2.2.1 – Programabilni logički kontroler DVP15MC11T



Deltin programabilni logički kontroler, prikazan na slici 2.2.1, punog naziva i oznake glasi **DVP15MC11T**. Iz oznake možemo iščitati kako se radi o PLC-u iz serije DVP-MC koji za napajanje koristi istosmjernu struju od 24 V, dok mu je izlaz tranzistorskog tipa. Ovaj PLC može upravljati do 24 realnih osi, a korištenjem virtualnih može upravljati do 32 osi. Ima ugrađene naredbe za kontrolu pokreta robota, tj. osi, i to ga čini jednostavnim za korištenje. PLC sadrži brzi procesor koji radi na frekvenciji od 1 GHz i može obavljati sinkronizaciju osi u vremenu od 2 ms, za 4 osi, i vremenu od 4 ms, za 8 osi. Kapacitet programa koje možemo isprogramirati na PLC-u iznosi 20 MB, dok memorijski kapacitet za varijable iznosi isto 20MB. Ovaj PLC ima prednost jer podržava korištenje, tj. ima mogućnosti čitanja i izvršavanja naredbi G-koda. Jedan G-kod program na PLC-u može imati veličinu do 256 KB, a istovremeno može biti do 64 takvih G-kod programa. Računalno sučelje u kojem se programira ovaj PLC je program *CANopen builder*.

Komunikacija između PLC-a i računala vrši se preko komunikacijskog priključka Ethernet, dok se komunikacija PLC-a i servo pogona (eng. Servo drive) izvodi preko CANopen priključka za gibanje (eng. CANopen Motion) korištenjem komunikacijskog protokola CANopen DS402.

Ethernet je tradicionalna tehnologija za povezivanje uređaja u žičanu lokalnu mrežu (LAN) ili mrežu širokog područja (WAN). Omogućuje uređajima da međusobno komuniciraju putem protokola. Sustavi koji komuniciraju preko Etherneta dijele tok podataka u kraće dijelove koji se nazivaju poruke. Svaka poruka sadrži adresu izvora i destinacije i podatke za provjeru grešaka kako bi se oštećene poruke mogle otkriti i odbaciti. Najčešće protokoli višeg sloja pokreću ponovni prijenos izgubljenih poruka.

CAN (eng. Controller area network) je robustan standard sabirnice koji omogućuje mikrokontrolerima i uređajima da međusobno komuniciraju bez posredovanja glavnog računala. To je protokol temeljen sa slanju poruka i dizajniran je za višestruko električno ožičenje unutar nekog sustava. Za svaki uređaj, podaci se šalju u porukama koje se onda prenose serijski, ali na takav način da ako više od jednog uređaja šalje u isto vrijeme, uređaj s najvišim prioritetom može nastaviti dok ostali odustaju.

CANopen je komunikacijski protokol i specifikacija profila uređaja za sustave koji se koriste u automatizaciji. CANopen standard sastoji se od sheme adresiranja, nekoliko malih komunikacijskih protokola i aplikacijskog sloja definiranog profilom uređaja. Komunikacijski protokoli imaju podršku za upravljanje mrežom, nadzor uređaja i komunikaciju između čvorova, uključujući jednostavan prijenosni sloj za segmentaciju/desegmentaciju poruka. Protokol niže razine koji implementira podatkovnu vezu i fizičke slojeve obično je CAN (eng. Controller area network), iako uređaji koji koriste neka druga sredstva komunikacije (kao što je Ethernet Powerlink, EtherCAT) također mogu implementirati CANopen profil uređaja.

Maksimalna brzina komunikacije iznosi 1 Mbps, a maksimalna udaljenost na kojoj se komunikacija može izvoditi iznosi 100 m.

Osim komunikacijskih priključaka Ethernet i CANopen, PLC sadrži još priključak za RS-232 i RS-485 komunikaciju, a uz modul za proširivanje PLC-a može se dobiti priključak za

ProfiBus komunikaciju. ProfiBus je otvoreni komunikacijski standard koji se koristi za serijsku komunikaciju nadređenih i/ili podređenih jedinica spojenih na zajedničku sabirnicu.

PLC još sadrži 16 ulaznih priključaka i 8 izlaznih. Tu su još i priključak za 24 voltno napajanje i utor za SD karticu.

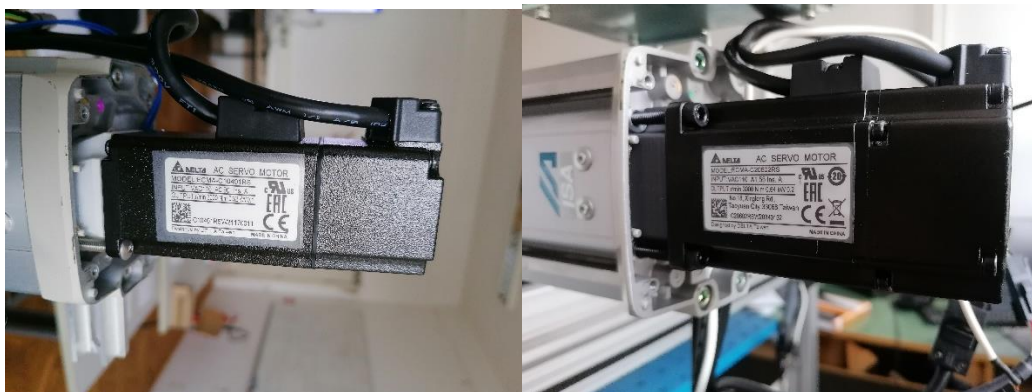
Osim PLC-a, postoje još dva dijela opreme pomoću kojih ostvarujemo mogućnost upravljanja i kontroliranja kartezijevog robota, a to su: servo pogoni (eng. Servo drives) i motori. Servo pogoni dolaze iz Deltine serije pogona ASDA-A2 i za svaku os se koristi jedan. Pogoni (eng. Drives) su međusobno povezani stvarajući lanac u komunikaciji, dok je samo jedan pogon (eng. Drive) povezan preko CANopen priključka s PLC-om. Oznake pogona koji se koriste glase: **ASD-A2-0121-M** i **ASD-A2-0221-M**. Prema oznaci možemo vidjeti kako se radi o servo pogonima iz serije A2, koji na izlazu daju snagu 100 W, odnosno 200 W, čije ulazno napajanje iznosi 220 V i da su modeli koji podržavaju komunikaciju preko mreža RS-485 i CANopen. Pogoni koji daju veću snagu stavljeni su na osi X i Y, jer se na njima koriste motori većih snaga, a pogoni koji daju manju snagu su stavljeni na osi Z i R. Na slici 2.2.2 su prikazani servo pogoni koji se koriste.



**Slika 2.2.2 - Servo pogoni snaga 200 W (lijevo) i 100 W (desno)**

Servo motori koji se koriste dolaze iz Deltine ECMA serije motora. ECMA motori mogu sadržavati inkrementalni enkoder, s 20-bitnom ili 17-bitnom rezolucijom, ili apsolutni enkoder, sa 17-bitnom rezolucijom, koji omogućavaju nesmetan i kontinuiran rad motora i povećavaju preciznost pri pozicioniranju osi. Frekvencija reakcije motora iznosi do 1 kHz,

vrijeme zaustavljanja motora je manje od 1 ms, a vrijeme za postizanje tražene brzine motora, za okretaje u rasponu od  $-3000 \frac{r}{min}$  do  $3000 \frac{r}{min}$ , iznosi 7 ms. Pune oznake servo motora glase: **ECMA-C20602RS** i **ECMA-C10401RS**. Prema oznakama možemo iščitati kako je riječ o servo motorima čije napajanje iznosi 220 V, a najveća moguća brzina vrtnje iznosi  $3000 \frac{r}{min}$ . Motori koji koriste 20-bitni inkrementalni enkoder (**C1**), odnosno 17-bitni inkrementalni enkoder (**C2**). Na slici 2.2.3 su prikazani modeli servo motora.



**Slika 2.2.3 - Servo motori snaga 100 W (lijevo) i 200 W (desno)**

Osim glavnih komponenti, bez kojih rad robota ne bi bio moguć, na robotu se koriste dodatna oprema koja omogućuje robotu, ali i korisniku, precizniji, bolji i brži rad. Od opreme tu mogu biti senzori, spojke, držači, distanceri itd. Senzori koji se koriste na našem kartezijevom koordinatnom robotu su krajnji prekidač (eng. End switch) i induktivni senzor.

Krajnji prekidač (eng. End switch) je postavljen na svaku os kako bi signalizirao krajnji položaj osi. Na ovom robotu takvi senzori su postavljeni na samo jednoj strani, tamo gdje je početni položaj osi (koordinata iznosi 0). Njihova funkcija je omogućavanje provedbe inicijalizacije osi pri svakom paljenju robota tijekom koje robot mora doći u isti početni položaj gdje mu koordinate iznose 0, 0, 0. Upravo zato da bi se svaki put tijekom inicijalizacije postiglo da se robot postavi u identični početni položaj, koriste se senzori krajnjeg položaja.

Induktivni senzor je postavljen na R-osi, točnije na hvataljki robota. Njegova funkcija je signaliziranje početnog položaja u  $0^\circ$  odnosno krajnjeg u  $360^\circ$ , pošto je riječ o istom položaju. Rad R-osi je takav da os ima raspon od  $0^\circ$  do  $360^\circ$ , ali nije u mogućnosti raditi takozvane krugove. Znači kada u jednom smjeru rotacije dođe do krajnjeg položaja više ne može nastaviti tim smjerom vrtnje, nego mora promijeniti smjer rotacije.

## 2.3 Usporedba postojeće i odabrane opreme

Usporedbu trenutne i nove opreme možete vidjeti u tablici 2.3.1.

	Oprema	
	IAI (eng. International Automation Industry )	Delta Electronics
<b>Kontroler</b>	IAI X-SEL-K-4-150IB-60I-60IB-30RIL-N1-EEE-5-1	DVP15MC11T
<b>Napajanje kontrolera</b>	220 V	24 V
<b>Broj osi za upravljanje</b>	1 do 4	24 realne, 32 virtualne 32 realne + virtualne
<b>Programi</b>	Do 64 programa	20MB memorije za progame 20 MB memorije za varijable
<b>G-kod</b>	Ne podržava	Do 64 programa Jedan program do 264 KB
<b>Računalni softver</b>	IA-101-X-MW	Program CANopen builder
<b>Komunikacijski priključci</b>	DeviceNet, CC-link, ProfiBus, Ethernet	Ethernet, CANopen, EtherCAT, DeviceNet, ProfiBus
<b>Broj ulaza</b>	32	16
<b>Broj izlaza</b>	16	8
<b>Sinkronizacija osi</b>	2 osi	4 osi (2 ms) 8 osi (4 ms)
<b>Snage motora</b>	150 W – X-os 60 W – Y i Z-os 30 W – R-os	200 W – X i Y-os 100 W – Z i R-os

**Tablica 2.3.1 - Usporedba trenutne i nove opreme**

Zbog zastarjelosti opreme kompanije IAI (eng. International Automation Industry ), koja se na ovom kartezijevom robotu koristi preko 20 godina, i zbog toga što se njihovo korisničko sučelje, preko kojeg se programira i upravlja robotom, sve manje koristi u svijetu automatizacije, odlučili smo se za promjenu upravljačke opreme robota.

Pri promjeni opreme na kartezijevom koordinatnom robotu zamijenio se je kontroler, zaslužan za upravljanje robotom, i korisničko sučelje preko kojeg se robot programira. Umjesto postojećeg kontrolera **IAI X-SEL-K-4-150IB-60I-60IB-30RIL-N1-EEE-5-1** izabran je programabilni logički kontroler **DVP15MC11T**. Osim promjene kontrolera došlo je do zamjene motora na osima i stavljanje dodatnog dijela opreme, dio koji je integriran u dosadašnjem kontroleru dok je u sklopu nove opreme on zaseban element, servo pogoni (eng. Servo drives), za svaku os po jedan. Na osima su izabrani jači motori, pa je tako na osi X došlo do promjene motora snage 150 W za motor snage 200 W. Motor iste snage je stavljen na os Y, zamjenjujući tako motor snage 60 W, a na osi Z i R su postavljeni motori snage 100 W koji su zamijenili njihove motore snage 60 W na Z-osi, odnosno motor snage 30 W na R-osi. Aktuatori serije ISA/ISPA, odnosno njihove baze, linearne vodilice i kuglični vijci ostali su nepromijenjeni i nastavit će se koristiti s novom upravljačkom opremom na kartezijevom koordinatnom robotu.

### 3. Sklapanje svih komponenti robota i elektroničke sheme spajanja

U prethodnim poglavljima, 2.2 i 2.3, opisane su nove komponente kartezijskog koordinatnog robota s kojima je robot dobio novu i veću dimenziju u svome radu. Opisane su komponente čija je promjena, u odnosu na staru opremu, donijela određenu promjenu u radu samog robota, dok zbog uvođenja nekih novih komponenti, kao novi i jači motori na osima, morale su se mijenjati komponente koje nisu mijenjale samu funkcionalnost rada robota, npr. spojke.

U ovom poglavlju bit će te upoznati sa svim komponentama koje čine našeg kartezijskog koordinatnog robota i bit će prikazane i opisane sve sheme elektroničkog spajanja komponenti robota.

#### 3.1 Komponente robota

Popis svih novih i starih komponenti sklopljenih i stavljenih na robot možete vidjeti u tablici 3.1.1.

	Količina
PLC - DVP15MC11T	1
Servo pogon - ASDA-A2	4
Motor – 200W	2
Motor – 100W	2
Spojka	4
Distancer	3
Prirubnica	1
Kučište motora	3
End switch	3
Indukcijski senzor	1
Kočnica	1
Aktuator - ISA/ISPA	3

**Tablica 3.1.1 – Popis opreme kartezijskog koordinatnog robota**

Sliku kartezijskog robota o kojem se priča u ovom radu možete vidjeti u prvom poglavlju na slici 1.1, no, detaljniji prikaz i opis svakog dijela robota slijedi u nastavku.

Kao što smo već opisali, kartezijski robot se sastoji od tri translacijske osi koje se nalaze u međusobnom odnosu gdje je svaka os pod pravim kutem u odnosu na preostale dvije osi. Okvir samog robota čine standardizirani profili i na donjoj plohi radnog prostora robota se nalazi plavi stol. Izgled cijelog robota možete vidjeti na slici 3.1.2.



**Slika 3.1.2 – Prikaz cijelog kartezijskog koordinatnog robota**

Aktuator X-osi se nalazi na samom standardiziranom profilu, dok su aktuatori preostalih dviju osi postavljeni na aktuatoru prethodne osi, rotacijska os R je ukomponirana zajedno s Z-osi. Dimenzije svake osi su:

- X-os: 600mm
- Y-os: 400mm
- Z-os: 100mm
- R-os: 360°

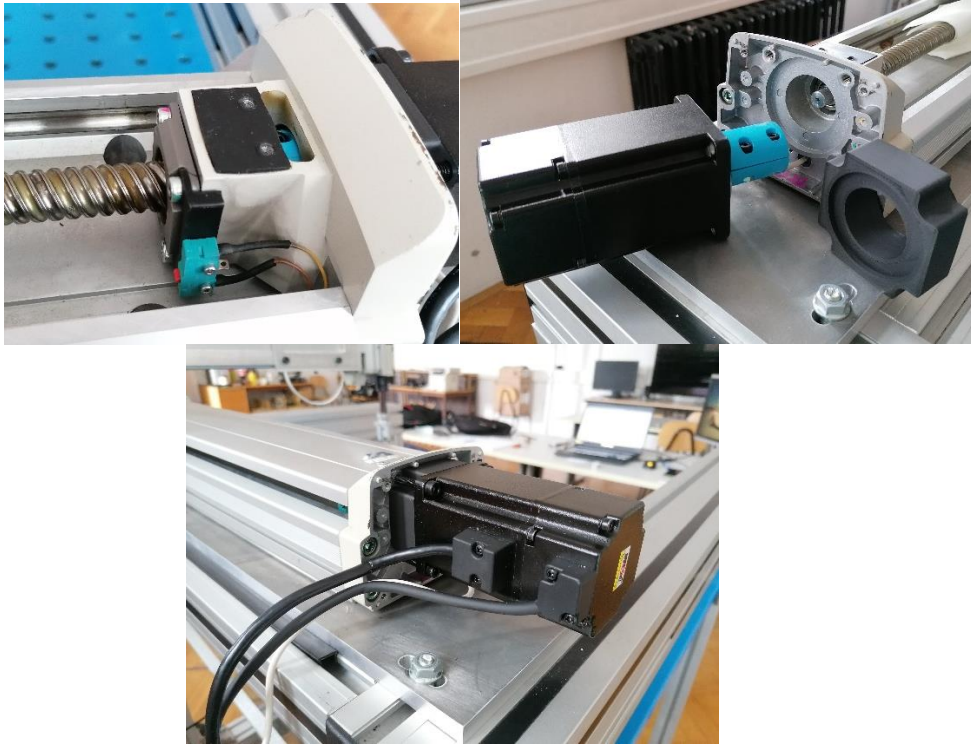
Funkcionalno, preinake koje su se radile na osima kartezijskog robota su za svaku os iste, ali prilagođene toj osi. Na primjeru X-osi ćemo prikazati sve promjene koje su napravljene. Pa tako, zbog stavljanja novog motora na aktuatoru su se morale staviti sljedeće nove komponente:

- Motor snage 200W
- Kućište motora



- Distancer
- Spojka
- End switch

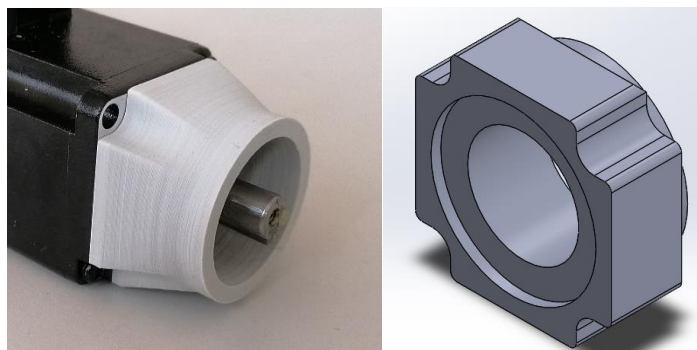
Na slikama 3.1.3 možete vidjeti sklop navedenih komponenti na osi X.



**Slika 3.1.3 – Sklop komponenti na aktuatoru X-osi**

Na principu sklapanja komponenti na osi X su sklopljene i komponente na ostalim osima, ali naravno prilagođene dimenzijama motora i vratila koji su stavljeni na svaku os.

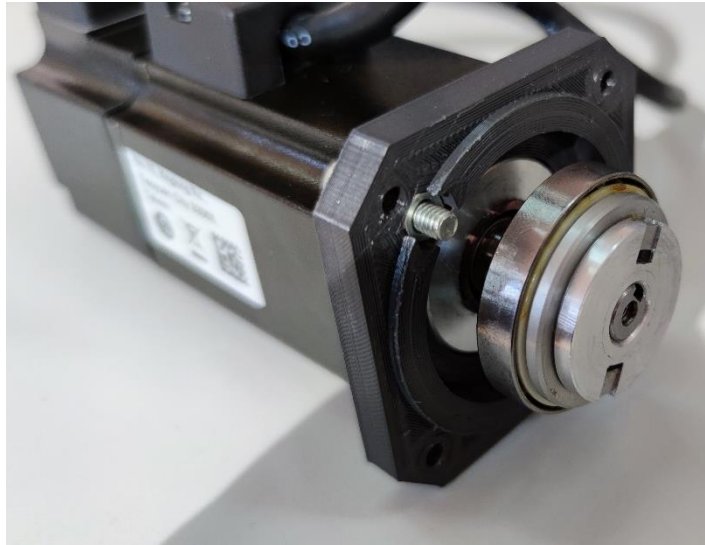
Zbog različitih dimenzija i snaga motora na osima, na slikama 3.1.4 možete vidjeti prikaz distancera za svaku os.



**Slika 3.1.4 – Distancer Z-osi (lijevo) i distancer X i Y-osi (desno)**

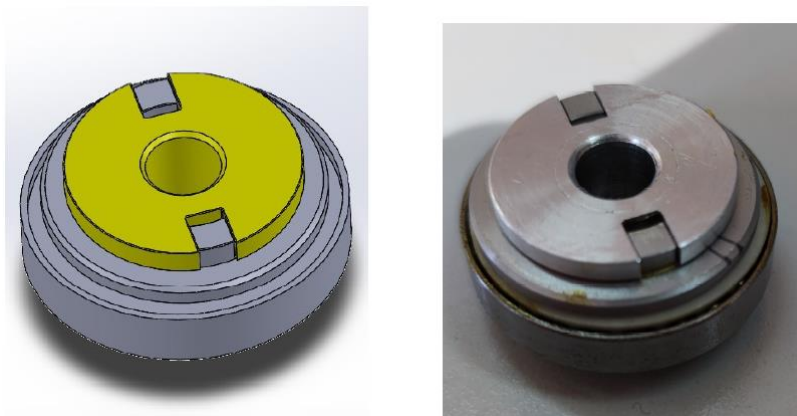
Kod rotacijske osi R dolazi do promjene. Kod osi X, Y i Z su stavljena kućišta motora kako bi se sklop vratila, spojke i kugličnog vretena zaštitio od vanjskih utjecaja, dok kod osi R nije bilo potrebe za tim jer je umjesto distancera i kućišta izrađena prirubnica koja uz svoju primarnu funkciju obavlja i funkciju zaštite sklopa vratila, spojke i harmonijskog prijenosnika. Osim toga na rotacijskoj osi nije bilo potrebe za postavljanjem krajnjeg prekidača (eng. end switch) zato što se umjesto njega za određivanje početnog položaja koristi indukcijski senzor kao sastavni dio hvataljke robota na vrhu osi.

Prikaz sklopa komponenti na osi R možete vidjeti na slici 3.1.5.



Slika 3.1.5 – Sklop komponenti na R-osi

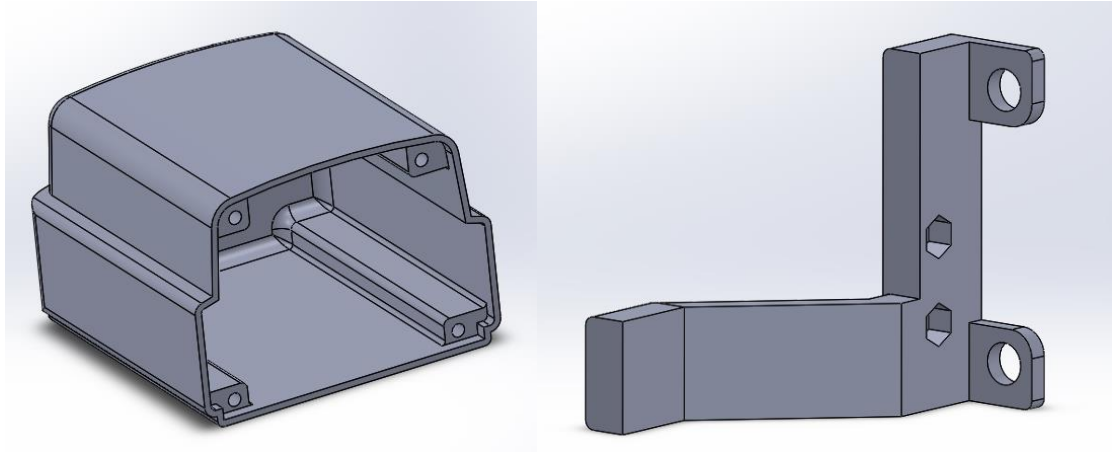
Na rotacijskoj osi R dolazi do promjene i kod samog aktuatora pokreta u odnosu na X, Y i Z osi. Umjesto kugličnog vretena koristi se harmonijski prijenosnik s prijenosnim omjerom 1:25. Zbog promjene motora na osi R morala se je izraditi spojka prilagođena prijenosu momenta sa servo motora **ECMA-C10401RS** na harmonijski prijenosnik. Izgled sklopa spojke i prijenosnika možete vidjeti na slici 3.1.6.



Slika 3.1.6 – Sklop spojke i harmonijskog prijenosnika



Kao što ste mogli primijetiti na prethodnim slikama, komponente poput kućišta motora, distancera, spojki, prirubnice i držača krajnjih prekidača su izrađene aditivnom tehnologijom 3D ispisa. Pošto je riječ o robotu koji će se koristiti u nastavne svrhe gdje neće biti izložen velikim i učestalim opterećenjima neke komponente ostat će nepromijenjene, dok komponente od iznimne važnosti za rad samog robota, spojke, su u kasnijoj fazi rada na robotu posebno izrađene od aluminija i onda su zamijenile svoje replike koje su bile napravljene postupkom 3D ispisa.



**Slika 3.1.7 – CAD prikaz kućišta motora (lijevo) i držača krajnjeg prekidača (desno) za X-os**

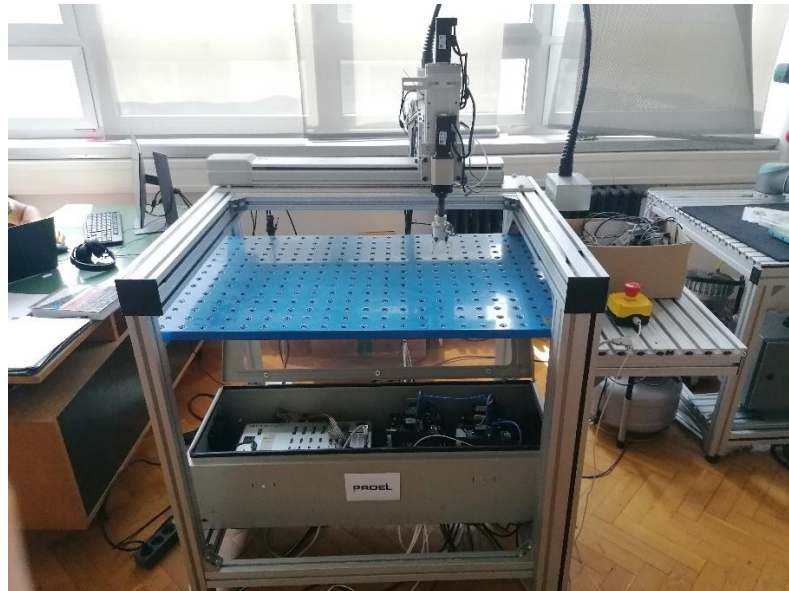
Radioničke crteže držača krajnjeg prekidača, distancera i kućišta za X-os možete vidjeti u prilogima (VI.).

### **3.2 Elektroničke sheme spajanja**

U ovom poglavlju opisat će se i prikazati način elektroničkog spajanja komponenti kartezijskog robota. Popis svih komponenti za spajanje:

- Programabilni logički kontroler (PLC)
- Servo pogon za X, Y, Z i R-os
- Motor za X, Y, Z i R-os
- Krajnji prekidač (eng. End switch) za X, Y i Z-os
- Indukcijski senzor za R-os
- Kočnica Z-osi
- Prekidač za hitne slučajeve (eng. Emergency button)
- Sabirnica

Sustav elektronike kartezijskog robota je osmišljen tako da su komponente postavljene na dva položaja, jedan unutar elektro ormara koji se nalazi ispod samog robota i drugi koji je na aktuatorima robota. Prikaz položaja i izgleda elektro ormara možete vidjeti na slici 3.2.1.



**Slika 3.2.1 – Položaj elektro ormara**

Komponente postavljene unutar ormara su PLC, servo pogoni i sabirnica. Elektro ormar je preko svojih konektora povezan direktno na napajanje, a preko tih konektora i pretvarača napona šalje napajanje do PLC-a i sabirnice. Osim novih komponenti unutar ormara se nalazi i zastarjeli kontroler koji je u potpunosti od spojen i trenutno se više ne koristi. Raspored komponenata unutar ormara možete vidjeti na slici 3.2.2.



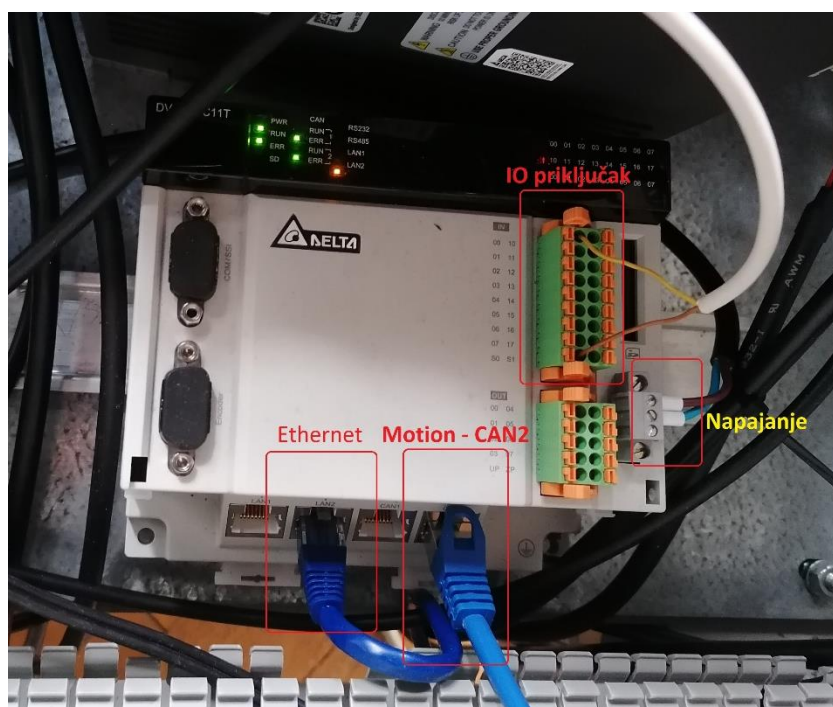
**Slika 3.2.2 – Elektro ormar**

Ostale komponente, motori, krajnji prekidači, indukcijski senzor i kočnica, se nalaze na aktuatorima osi robota, a prekidač za hitne slučajeve se nalazi na vidljivom i dostupnom mjestu na postolju pored robota.

## Programabilni logički kontroler – PLC

Programabilni logički kontroler DVP15MC11T preko svojih priključaka povezan je na sljedeće načine:

- Preko naponskog pretvarača dobiva napajanje od 24 V
- Komunikaciju s upravljačkim računalom ostvaruje preko ethernet priključka i ethernet kabela
- Komunikaciju sa servo pogonima ostvaruje preko Motion – CAN2 priključka pomoću ethernet kabela
- Preko svojih ulazno-izlaznih (eng. Input-output) priključaka poveza s prekidačem za hitne slučajeve

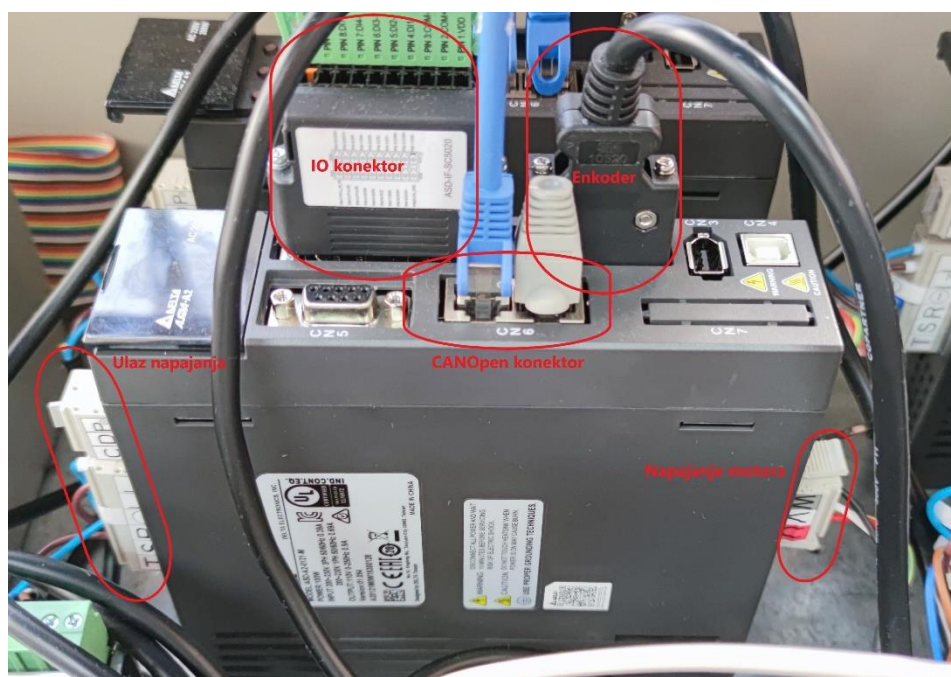


Slika 3.2.3 – Prikaz priključaka na PLC-u

## Servo pogoni

Postupak spajanja svakog servo pogona je proveden jednako, jedino je razlika kod servo pogona za Z-os gdje na ulazno-izlaznim priključcima imamo spojenu kočnicu koja je potrebna za držanje osi kako bi se spriječilo propadanje osi kada na motoru nestane napajanje. Servo pogoni su preko svojih priključaka povezani na sljedeći način:

- Na ulazni priključak dolazi napajanje od 240 V
- Na izlazu daje trofazni izmjenični napon od 400 V i preko njega je povezan s motorom na aktuatoru osi
- Na servo pogon dolazi signal s enkodera pomoću posebnog kabela enkodera motora
- Komunikacija između servo pogona i kontrolera izvedena je ethernet kabelom preko CANopen priključka pomoću CANopen protokola
- Preko ulazno-izlaznih priključaka povezan je s napajanjem od 24 V, prekidačem za hitne slučajeve, krajnjim prekidačem (odnosno indukcijskim senzorom za R-os) i, samo za slučaj Z-osi, s kočnicom



Slika 3.2.4 – Prikaz priključaka na servo pogonu

## Motori

Svaki motor koji se koristi na osima kartezijskog robota dolazi s dva priključka odnosno 2 kabela:

- Priključak/kabel motora
- Priključak/kabel enkodera

Postupak elektroničkog spajanja komponenti podijelili smo na dvije sheme:

- Blok shema spoja PLC-a, servo pogona i servo motora (I.)
- Shema spajanja kontakata PLC-a i servo pogona s ostalim komponentama (II.)

Sheme spajanja možete vidjeti na kraju rada u prilogu.

## 4. PLC program za upravljanje robota

Nakon odabira nove upravljačke opreme (poglavlje 2.2), sklapanja robota s novim i starim komponentama i spajanja elektronike svih komponenata (poglavlje 3.), čime dolazimo do kraja hardverske rekonstrukcije robota, slijedi izrada programa za programabilni logički kontroler (PLC) kako bi se robot mogao početi koristiti tj. kako bi se mogle početi ostvarivati željene kretnje našeg kartezijevog koordinatnog robota.

Programsko rješenje je podijeljeno u dva dijela, programski kod i python skripta. Programski kod unutar PLC-a izrađen je u ljestvičastom dijagramu (Ladder diagram) sa svrhom direktne kontrole pojedinih osi prema naredbama dobivenim s računala. Računalo naredbe generira pomoću python skripte s grafičkim sučeljem kroz upravljački program također napisan u programskom jeziku python. Svrha izrade zasebnog upravljačkog programa je veća fleksibilnost rada sustava te mogućnost implementacije postojeće kontrole u neki komercijalno dostupan program (RoboDK).

### 4.1 Programski kod PLC-a (Ladder diagram)

Programski kod PLC-a je izrađen u programskom softveru CANopen Builder. CANopen Builder je softver za implementiranje programskih kodova u programabilne logičke kontrolere i proizvod je firme Delta. Razlog odabira ovog softvera je bio mogućnost konfiguracije svakog servo pogona zasebno i mogućnost rada s njima kroz jedno sučelje.

Svrha izrade programa je kako bi se omogućila direktna kontrola svih osi robota i kako bi se definiralo gibanje svake pojedine osi. Trenutno se koristi indirektna kontrola robota preko python skripte, no kod je napisan na takav način da je moguće implementirati fizičku kontrolu na robotu, što bi na primjer bilo korištenje privjeska za upravljanje (eng. Teach pendant).

U našem slučaju programski kod implementiran na PLC-u izrađen je u ljestvičastom dijagramu (eng. Ladder diagram). Sami dijagram napisan je po sekvencijalnoj metodi izvođenja i podijeljen je od početka, od paljenja osi i inicijalizacije svake osi, do grupiranja i pokretanja grupe osi. Koriste se funkcijski blokovi s kojima se definira način gibanja osi.

Prije početka pisanja programskoga koda, potrebno je definirati značajke svake osi robota (vrsta osi, prijenosni mehanizam, iskoristiva duljina osi i korak navoja). U tablici 4.1.1 možete vidjeti definirane značajke svake osi.

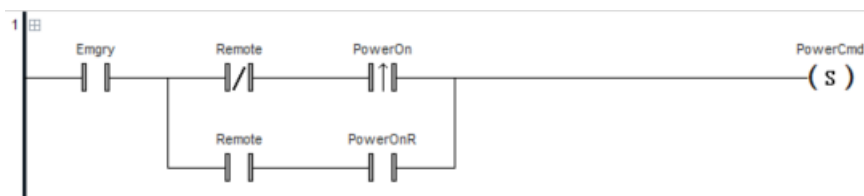


	<b>X</b>	<b>Y</b>	<b>Z</b>	<b>R</b>
Vrsta osi	Linearna	Linearna	Linearna	Rotacijska
Prijenosni mehanizam	Kuglični vijak	Kuglični vijak	Kuglični vijak	Harmonijski prijenosnik
Iskoristiva veličina	590 mm	390 mm	90 mm	360°
Korak navoja	20 mm	20 mm	4 mm	-

**Tablica 4.1.1 – Značajke osi X, Y, Z i R**

Nakon definiranja značajki osi slijedi djelomični prikaz koda uz objašnjenje postignutih funkcija i prikazat će se tehnički podaci za svaku os. Kompletan programski kod možete vidjeti na kraju rada u prilogu (III.).

Kao što smo već rekli, ljestvičast dijagram (eng. Ladder diagram) se izvršava sekvencijalno (linija po linija) pa je tako prva linija koda ujedno i ona u kojoj se postiže početak (start) rada samog robota.

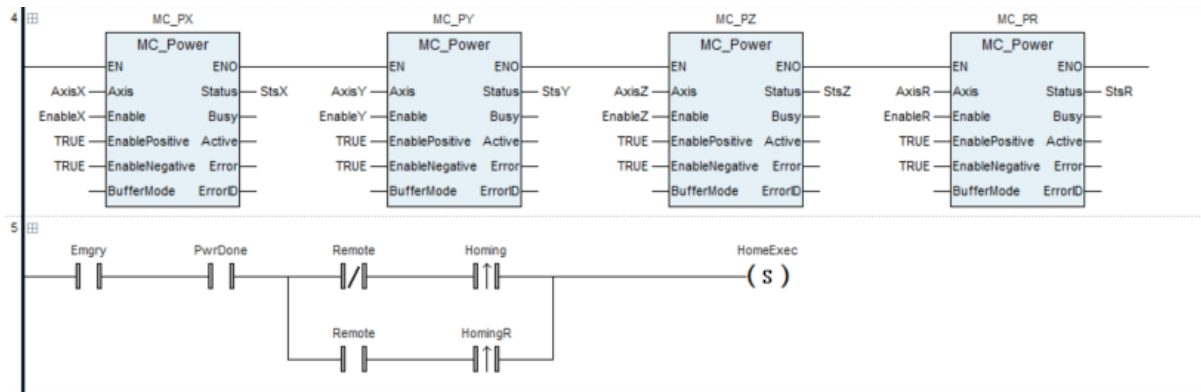


**Slika 4.1.2 – Programski kod, Linija 1**

Prekidač za hitne slučajeve, signal Emgry, postavljen je kao obični radni kontakt, ali njegovo djelovanje je drugačije u odnosu na normalne prekidače. Dok eng. Emergency button nije pritisnut njegovi kontakti su spojeni što će u Ladder dijagramu rezultirati propuštanjem signala. Smisao je da, u slučaju pritiskanja prekidača za hitne slučajeve, on od spoji svoje kontakte i samim time prekine propuštanje signala i dovede do zaustavljanja svih osi robota.

Osim njega postavljeni su i kontakti Remote i PowerOn. Napravljena je logika koja nam pokazuje i omogućuje spajanje na robot na 2 načina, direktno preko računala na PLC pomoću ethernet kabela ili bežično bez kabela (eng. Remote), ali nije moguće istovremeno spojiti se s robotom na oba načina. Takva logika ostvarena je pomoću kontakta Remote. Kontakt PowerOn propušta signal kada na robot dovedemo napajanje.

Nakon dovođenja napajanja, što se događa pri svakom inicijalnom paljenju robota, započinje se takozvano paljenje svake osi.



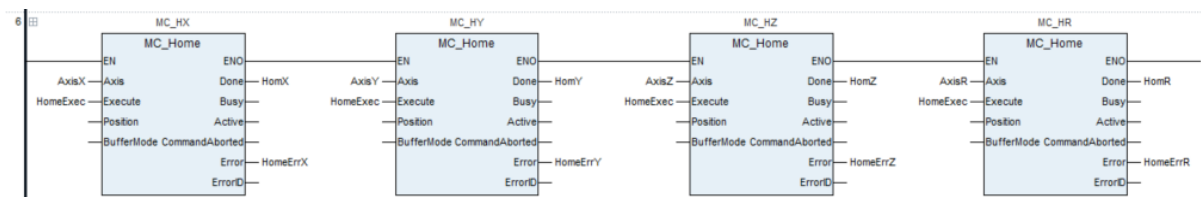
Slika 4.1.3 – Programski kod, Linije 4 i 5

Statusni blok MC\_Power na svojem ulazu dobiva signale AxisX (Y, Z ili R), signale koji se aktiviraju nakon dovođenja napajanja, i na svom izlazu daje signal StsX (Y, Z ili R) koji označuju da je došlo do „paljenja“ svake osi, odnosno označavaju aktiviranje svake osi.

Nakon što su se sve osi inicijalno aktivirale signali StsX, StsY, StsZ i StsR preko statusnog bloka aktiviraju signal PwrDone.

Na početku, nakon inicijalnog paljenja robota i njegovih osi, potrebno je ostvariti radnju vraćanja osi u njihove početne položaje, takozvani eng. Homing. Kako bi se pokrenule osi i dovele u svoje početne položaje koristi se prekidač čiji naziv signala glasi Homing.

U međusobnom odnosu signala prekidača za hitne slučajeve Emgry, PwrDone, Remote i Homing, nakon pritiska prekidača za vraćanje osi u početni položaj aktivira se signal HomeExec i započinje proces Homing-a.



Slika 4.1.4 – Programski kod, Linija 6

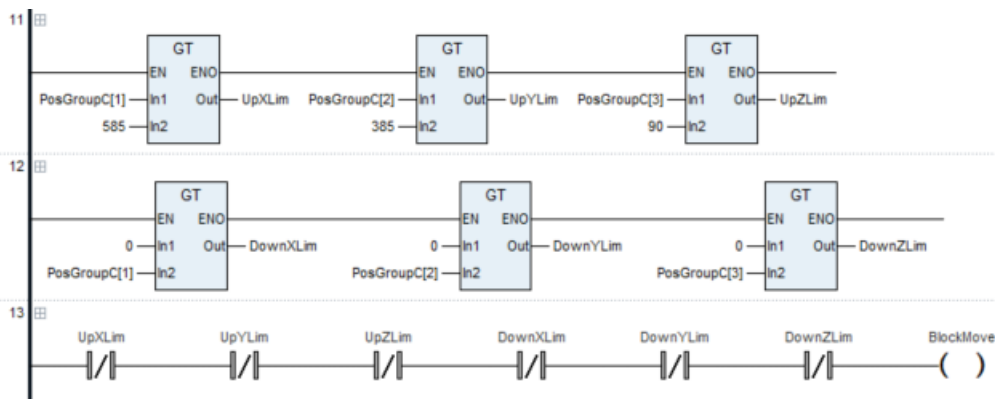
Blokovi MC\_Home na svojim ulazima primaju 2 signala, AxisX (Y, Z ili R) i HomeExec. Nakon što su oba dva aktivirana, paljenje osi i pritiskanje prekidača za Homing, svaka os započinje svoj postupak vraćanja u početni položaj.

Signalizacija da se je os vratila u početni položaj ostvaruje se aktiviranjem krajnjeg prekidača, tj. indukcijskog senzora za R-os, i nakon što se to dogodi blok MC\_Home na svojem izlazu daje signal HomX (Y, Z ili R) koji označavaju kraj radnje vraćanja osi u početni položaj.

Kasnije je kod napravljen tako da se pri početnom paljenju robota i pri inicijalizaciji osi mora obaviti radnja vraćanja osi u početni položaj, jer inače korisnik neće biti u mogućnosti slati naredbe na robota, tj naredbe neće ni doći do robota.

Nakon što nam je robot upaljen i sve njegove osi se nalaze u početnom položaju, možemo početi robotu zadavati naredbe. Kao što je prije rečeno, programsko rješenje upravljanja robota je podijeljeno u dva dijela. Pa je tako postupak zadavanja naredbi robotu ostvaren tako da se preko računala naredbe generiraju pomoću python skripte s grafičkim sučeljem. Detaljniji prikaz grafičkog sučelja slijedi u idućem poglavlju.

Nakon upisivanja i zadavanja naredbi tj. pomaka robotu, u Ladder dijagramu slijedi provjera konačnih koordinata točaka u koje robot treba doći.

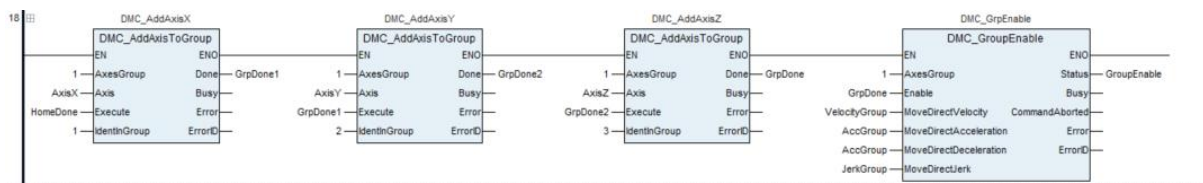


Slika 4.1.5 – Programski kod, Linije 11, 12, 13

Ukoliko jedna ili više koordinata budu izvan svojeg raspona, naredba tj pomak koji se je zadao neće se izvršiti i robot će javljati da je došlo do greške.

Nakon što je naredba prošla provjeru limita slijedi sama egzekucija pomaka od strane robota. Kako bi se kretanje robota maksimalno optimirale, da dobijemo direktne pomake od početne do krajnje točke, potrebno je postići sinkronizirano kretanje osi. To bi značilo da se sve osi istovremeno pokreću i istovremeno zaustavljaju, a ovisno o pomaku koji moraju ostvariti ubrzanje i brzina kretanja osi se prilagođavaju.

Ovakav rad osi se u Ladder dijagramu postiže postavljanjem osi u zajedničku grupu. Smisao postavljanja osi u grupu je i postizanje linearnog pomaka robota između dviju točki u prostoru.

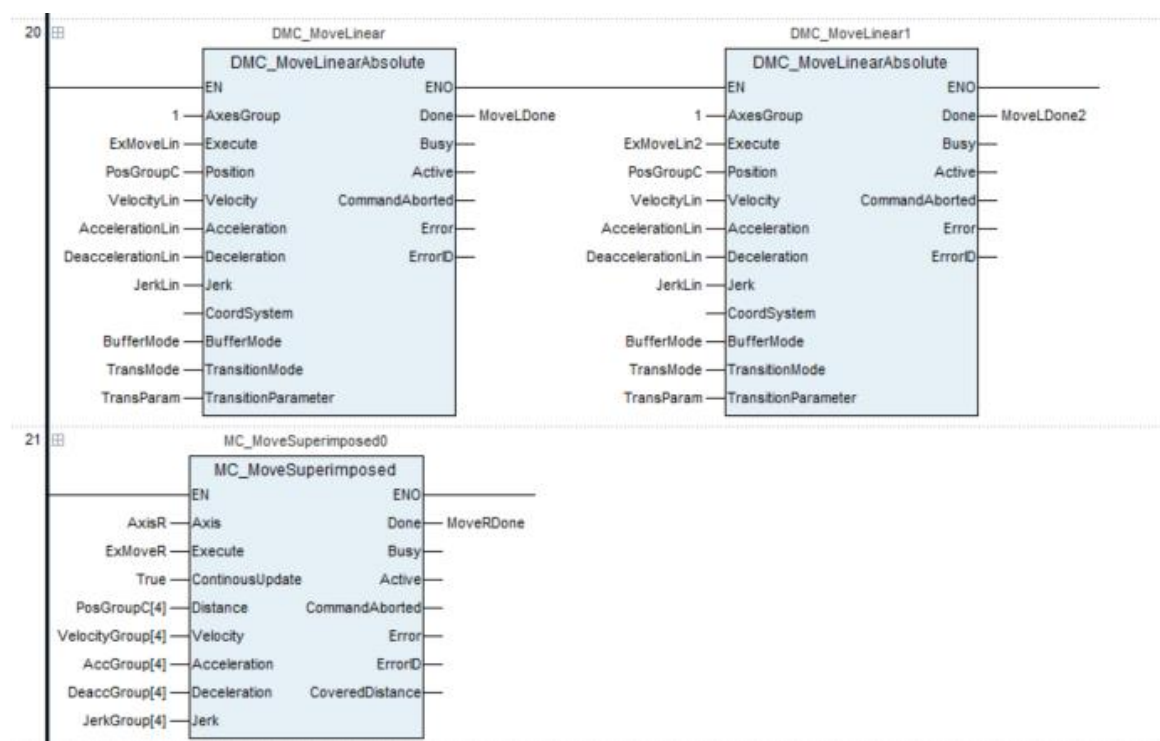


Slika 4.1.6 – Programski kod, Linija 18



Osim optimirane kretnje robota, sinkronizacija osi je potrebna kako bi bilo moguće ostvariti cirkularne pomake robota. U grupu osi nije uvrštena i os R. Razlog tome je hvataljka na samom vrhu osi i zbog nje su se htjele postići zasebne kretnje osi R koja nije u ovisnosti o kretanjima preostalih osi.

Postizanje linearnih i cirkularnih kretnji osi robota i zasebne rotacije R-osi, u odnosu na ostale, ostvaruje se blokovima DMC\_MoveLinearAbsolute, DMC\_MoveCircularAbsolute i MC\_MoveSuperimposed.



Slika 4.1.7 – Programski kod, Linije 20 i 21

Na ulazima blokova `DMC_MoveLinearAbsolute`, `DMC_MoveCircularAbsolute` i `MC_MoveSuperimposed` upisani su signali koji definiraju kinematiku gibanja osi robota.

Popis svih signala i varijabli mora se posebno definirati, od njihovih vrijednosti do memorijskih lokacija na koje se oni spremaju, i njega možete vidjeti na kraju rada u prilogu. Za potrebe definiranja kinematike gibanja osi robota, odnosno za definiranje ulaznih parametara blokova za linearno i cirkularno gibanje osi robota, definirane su sljedeće varijable:

	Veličina
VelocityLin i VelocityCirc	$50 \text{ mm/s}$
AccelerationLin i AccelerationCirc	$100 \text{ mm/s}^2$
DecelerationLin i DecelerationCirc	$100 \text{ mm/s}^2$
JerkLin i JerkCirc	$200 \text{ mm/s}^3$

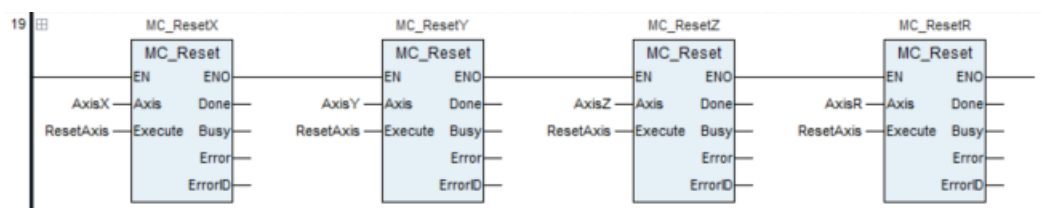
VelocityGroup	20 $mm/s$
AccelerationGroup	40 $mm/s^2$
DecelerationGroup	40 $mm/s^2$
JerkGroup	80 $mm/s^3$

**Tablica 4.1.8 – Popis varijabli za definiranje kinematike gibanja robota**

U tablici 4.1.2 možete vidjeti kako su vrijednosti za određivanje kinematike osi robota za linearno i cirkularno gibanje jednake. Blokovi za ta gibanja razlikuju se po tome što se bloku za cirkularno gibanje osim konačne pozicije zadaje i vrijednost radijusa kružnice, odnosno zadaje se položaj „vrha“ kružnog luka kroz koji bi robot trebao proći na putu do krajnje točke.

Osim vrijednosti za linearno i cirkularno gibanje tu su i vrijednosti za rotacijsko gibanje R-osi. Vrijednosti brzine, akceleracije, deceleracije i jerka se za rotacijsku os čitaju iz niza i metodom superponiranja definiraju kinematiku gibanja R-osi.

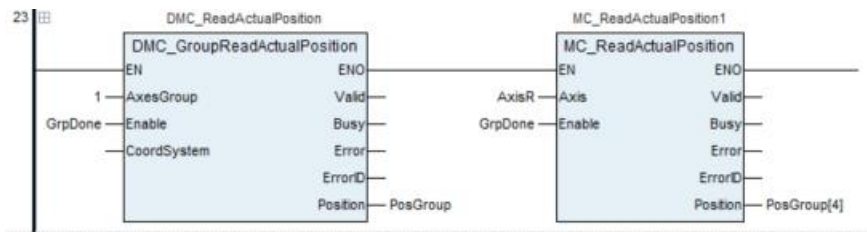
Korištenje kartezijskog robota, kao i bilo kojeg drugog robota, nakon programiranja nije idealno i često se zna dogoditi, osobito tijekom procesa razvoja robota, da se javljaju nekakve greške koje dovedu ili do zastoja rada robota ili do izvršavanja pogrešnih pomaka. Kako bi se robot mogao ponovo koristiti nakon pojave greške potrebno je ponovo odraditi inicijalizaciju povratka u početni položaj. Ta inicijalizacija se vrši pri svakom ponovnom paljenju robota, no, kako bi se izbjeglo konstantno paljenje i gašenje cijelog robota u programskom kodu stavljen je blok MC\_Reset za svaku os. Blok omogućuje takozvani blagi reset (eng. Soft reset) te nakon njegovog aktiviranja pokreće se postupak Homing-a.



**Slika 4.1.9 – Programski kod, Linija 19**

Na ulazu blokova MC\_Reset dolazi signal ResetAxis, što je ubiti signal prekidača za blagi reset koji nakon što se aktivira pokreće funkciju blokova MC\_Reset.

Za kraj programskog koda stavljeni su blokovi DMC\_GroupReadActualPosition i MC\_ReadActualPosition koji čitaju u svakom trenutku trenutnu poziciju, odnosno trenutne koordinate osi robota i na izlazu daju njihove vrijednosti.



Slika 4.1.10 – Programski kod, Linija 23

Preko python skripte pomoću ovih blokova, korisnik u svakom trenutku može vidjeti trenutne koordinate X, Y, Z i R osi robota.

## 5. Upravljanje robotom i primjer G-koda

U ovom poglavlju prikazat će se i objasniti grafičko sučelje za upravljanje robotom i priložit će se primjer G-koda za zadavanje naredbi robotu

Kao što je već rečeno u prethodnom poglavlju, programsko rješenje je podijeljeno u dva dijela, programski kod i python skripta. Programski kod za određivanje kinematike gibanja osi robota, opisan u prethodnom poglavlju, napisan je i postavljen u programabilni logički kontroler PLC. Napisan je radi direktne kontrole pojedinih osi prema naredbama dobivenim s računala. Računalo te naredbe generira pomoću python skripte s grafičkim sučeljem kroz upravljački program.

U ovom poglavlju neće se ulaziti u dubinu python skripti i posebno svaku objašnjavati, nego će se u kratkim crtama objasniti njihove funkcije.

Python skripte napisane su kako bi računalo moglo transformirati dobivene ulazne podatke od korisnika i tako transformirane slati ih na PLC radi izvršenja istih naredbi. Korisnik može zadati i/ili promijeniti sljedeće naredbe/podatke:

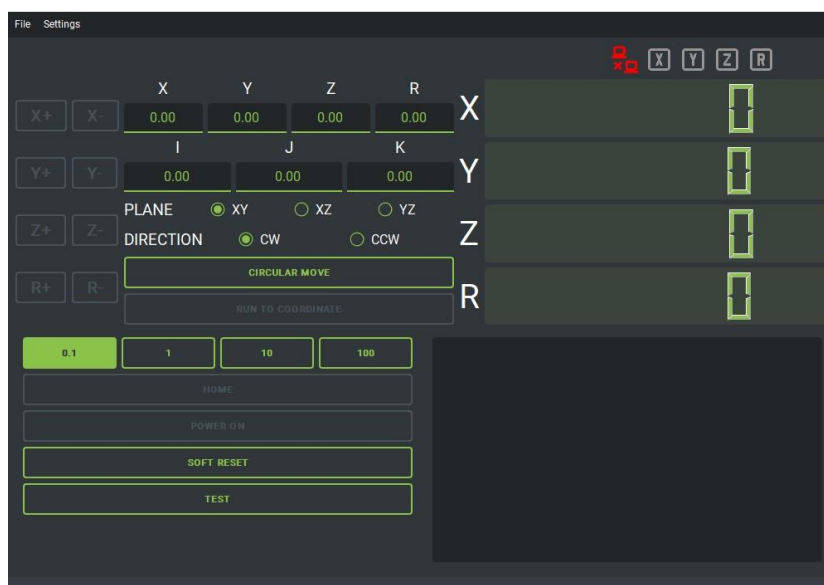
- Paljenje osi
- Inicijalizacija vraćanja u početni položaj osi
- Blagi reset osi
- Pomak po X, Y, Z i/ili R osi
- Koračni pomak svake osi zasebno
- Određivanje iznosa koračnog pomaka
- Linearno ili cirkularno gibanje
- Odabir ravnine za cirkularno gibanje
- Odabir smjera cirkularnog gibanja (u smjeru kazaljke na satu ili obrnuto)
- Veličinu kružnog luka cirkularnog gibanja
- Brzina (eng. Velocity)
- Akceleracija (eng. acceleration)
- Deceleracija (eng. deceleration)
- Jerk

Računalo povezano s PLC-om, pomoću ethernet kabela, dobiva naredbe od korisnika koje šalje na PLC.

Kako bi korisnik mogao upravljati robotom preko računala potrebno je bilo napraviti grafičko korisničko sučelje GUI (eng. graphical user interface). Grafičko sučelje napravljeno je i isprogramirano pomoću programa Qt Designer. Qt Designer je C++, JavaScript i QML integrirano razvojno okruženje za više platformi koje pojednostavljuje razvoj GUI aplikacija.

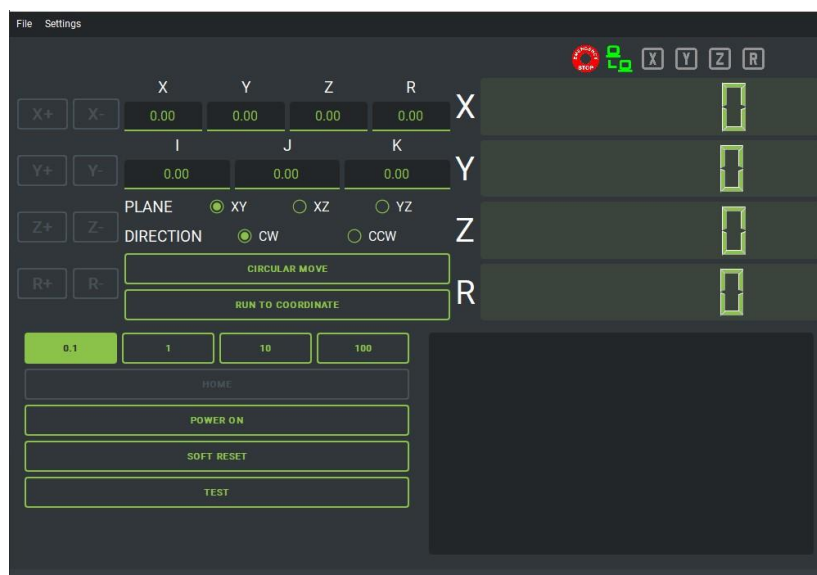
U nastavku slijedi prikaz grafičkog sučelja i opis kako se preko njega upravlja robotom.

Na slici 5.1 možete vidjeti prikaz početnog izgleda GUI prozora na samom početku korištenja robota, odnosno prije početka upravljanja s njime.



Slika 5.1 – Početni prozor grafičkog sučelja

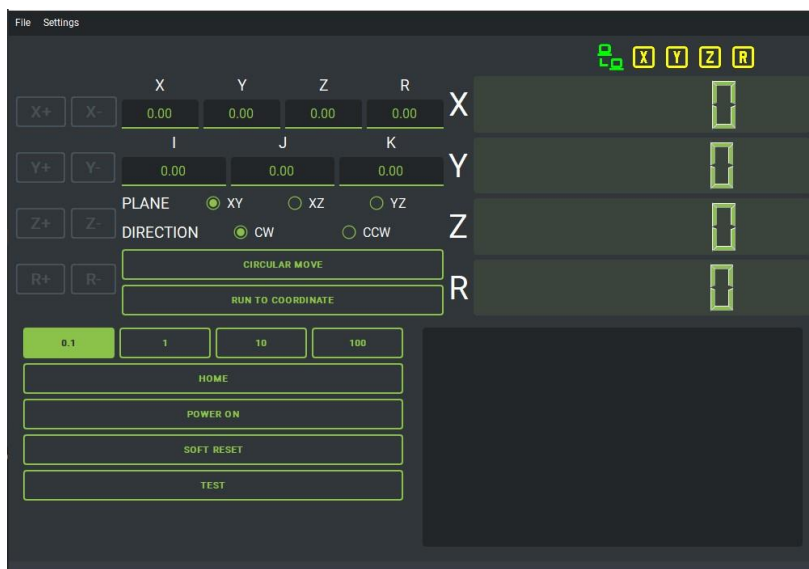
Nakon što smo fizički upalili robot, pustili napajanje u elektro ormar, i prije nego što možemo početi upravljati robotom, potrebno je povezati računalo s PLC-om. Pošto PLC niti u jednom trenutku nije „online“ i na njega se može spojiti samo pomoću ethernet kabela, potrebno je pri svakom ponovnom paljenju upisivati IP adresu računala kako bismo se mogli povezati s njim.



Slika 5.2 – Prozor grafičkog sučelja nakon povezivanja s PLC-om

Nakon što smo uspješno povezali računalo s PLC-om, možemo vidjeti promjenu boje ikone za povezanost računala i PLC-a, iz crvene u zelenu, i možemo vidjeti kako smo dobili mogućnost paljenja osi klikom na gumb „POWER ON“ (usporedbom slika 5.1 i 5.2). Omogućuje nam se početni postupak upravljanja robotom (opisan u poglavlju 4.1), paljenje osi i inicijalno vraćanje osi u početni položaj (eng. homing).

Nakon pritiska gumba „POWER ON“ dolazi do paljenja osi i omogućava nam se inicijalizacija osi (eng. homing).



**Slika 5.3 – Prozor grafičkog sučelja nakon paljenja osi**

Usporedbom slika 5.3 i 5.2 možemo vidjeti kako su ikone za status osi postale žute. Žuta boja označava dva slučaja:

1. Početna inicijalizacija nije izvršena
2. Došlo je do greške na servo pogonu osi

Isti izgled poprima prozor grafičkog sučelja prije početne inicijalizacije i nakon što se pojavi greška na servo pogonu neke osi.

Pritiskom gumba „HOME“ pokreće se vraćanje svake osi u njen početni položaj. Nakon aktiviranja krajnjeg prekidača signal s njega dolazi na PLC pa na računalo i ikona statusa osi poprima zelenu boju. Tek nakon što su sve osi postigle status odrađene inicijalizacije, korisnik može početi upravljati, tj. koristiti robota.

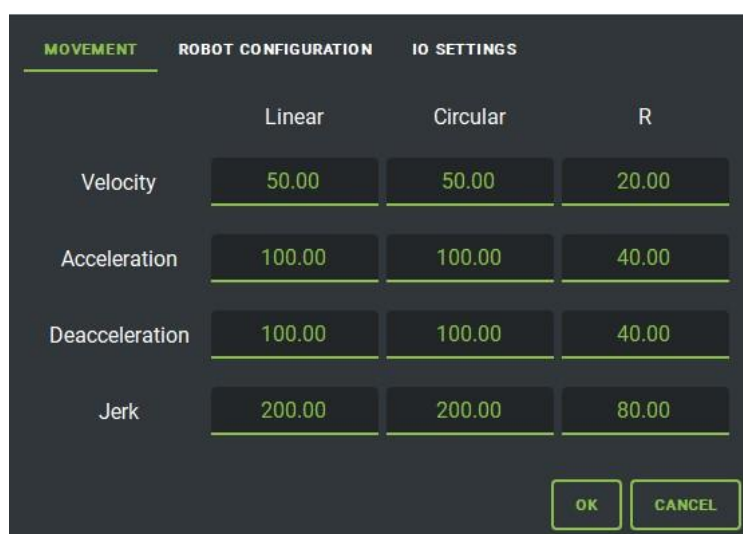
Na slici 5.4 možete vidjeti kako osim što su ikone za status osi postale zelene, omogućilo nam se koračno upravljanje svake osi zasebno (u lijevom gornjem kutu grafičkog sučelja). Osim što su nam sada svi gumbi, tj. sve funkcije, postale izvedive, na desnoj strani korisničkog sučelja možemo vidjeti ispis trenutnih koordinati svake osi. Takav ispis omogućen je korištenjem funkcijskog bloka statusa osi u programskog kodu PLC-a (opisano na kraju poglavlja 4.1) i povezivanjem grafičkog sučelja s tim vrijednostima pomoću python skripte.



Slika 5.4 – Prozor grafičkog sučelja tijekom upravljanja robotom

Osim koračnog upravljanja osim robota, korisniku je omogućeno i upisivanje svake koordinate i određivanja načina gibanja, „RUN TO COORDINATE“ (linearno gibanje) i „CIRCULAR MOVE“ (kružno gibanje), te je moguće upisati sve potrebne podatke za cirkularno gibanje robota.

Kako bi korisnik preko korisničkog sučelja mogao imati potpunu kontrolu nad samim robotom i mogao ga upravljati po parametrima potrebnim za izvršavanje određenih radnji, osim mogućnosti potpunog definiranja svakog pomaka moguće je i definirati varijable koje opisuju kinematiku gibanja robota.

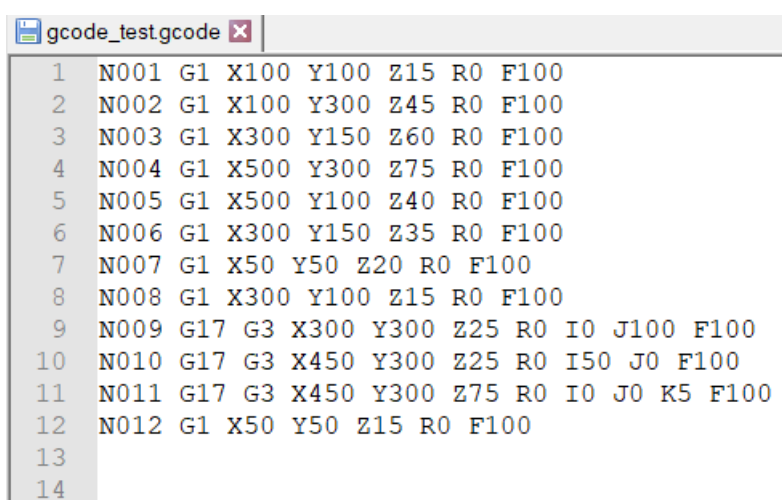


Slika 5.5 – Prozor grafičkog sučelja pri upisu varijabli

Pritiskom na gumb „Settings“ u gornjem lijevom kutu prozora, otvara nam se prozor grafičkog sučelja u kojem korisnik može upisati vrijednosti za četiri varijable koje definiraju kinematiku gibanja osi (slika 5.5). Brzina (eng. Velocity), akceleracija (eng. Acceleration), deceleracija (eng. Deceleration) i Jerk su vrijednosti koje definiraju linearno i cirkularno gibanje i mogu se zasebno upisati te iste varijable za gibanje R-osi.

Osim koračnog upravljanja robotom, zadavanja svakog pomaka posebno i određivanja radi li se o linearnom ili cirkularnom gibanju, napisana je posebna python skripta koja može transformirati G-kod u naredbe koje se šalju na PLC robota. Robot nakon toga izvrši pomake definirane G-kodom.

Python skripta čita naredbe G-koda iz tekstualne datoteke i šalje ih jednu po jednu na robota. Kod te skripte možete pročitati u prilogima (V.). Primjer G-koda možete vidjeti na slici 5.6.



```
gcode_test.gcode x
1 N001 G1 X100 Y100 Z15 R0 F100
2 N002 G1 X100 Y300 Z45 R0 F100
3 N003 G1 X300 Y150 Z60 R0 F100
4 N004 G1 X500 Y300 Z75 R0 F100
5 N005 G1 X500 Y100 Z40 R0 F100
6 N006 G1 X300 Y150 Z35 R0 F100
7 N007 G1 X50 Y50 Z20 R0 F100
8 N008 G1 X300 Y100 Z15 R0 F100
9 N009 G17 G3 X300 Y300 Z25 R0 I0 J100 F100
10 N010 G17 G3 X450 Y300 Z25 R0 I50 J0 F100
11 N011 G17 G3 X450 Y300 Z75 R0 I0 J0 K5 F100
12 N012 G1 X50 Y50 Z15 R0 F100
13
14
```

**Slika 5.6 – Primjer G-koda**

Ovime smo postigli automatizirano kretanje robota po putanjama koje želimo i koje smo napisali u G-kodu. Python skriptu koja služi za čitanje G-koda i slanje naredbi na robota možete vidjeti u prilogu.

Implementacijom programskog koda u PLC i povezivanjem računala s PLC-om pomoću napisanih python skripti, stvorio se temelj za daljnje korištenje robota.

Programskim kodom definirali smo kinematiku gibanja osi robota i način izvršavanja dobivenih naredbi, a pomoću python skripti omogućeno je da se u daljnjem razvoju ovog kartezijskog koordinatnog robota njega poveže s programima za programiranje robota poput RoboDK.



## 6. Zaključak

Kroz ovaj rad cilj je bio napraviti cjelovitu rekonstrukciju kartezijskog koordinatnog robota i omogućiti njegov daljnji rad u edukacijske svrhe te omogućiti daljnji razvoj samog robota. To se je i ostvarilo. Zamjena zastarjele opreme s novom te integracija nove opreme s ostatkom robota veliki su dio procesa rekonstrukcije. Potrebno je bilo izabrati kvalitetnu i modernu opremu koja prati današnje standarde za automatizaciju i robotiku u industriji i zbog toga je pala odluka na odabir i integraciju opreme firme Delta electronics. Zbog zastarjelosti opreme kompanije IAI (eng. International Automation Industry ), koja se na kartezijskom robotu, prikazanom u ovom radu, koristila preko 20 godina, i zbog toga što se njihovo korisničko sučelje, preko kojeg se programira i upravlja robotom, sve manje koristi u svijetu automatizacije, odlučili smo se za promjenu upravljačke opreme robota. Pri promjeni opreme na kartezijskom koordinatnom robotu zamijenio se je kontroler, zaslužan za upravljanje robotom, i korisničko sučelje preko kojeg se robot programira. Umjesto postojećeg kontrolera IAI X-SEL-K-4-150IB-60I-60IB-30RIL-N1-EEE-5-1 izabran je programabilni logički kontroler DVP15MC11T. Osim promjene kontrolera došlo je do zamjene motora na osima i stavljanje servo pogona (eng. Servo drives), za svaku os po jedan, dio koji je bio integriran u prijašnjem kontroleru dok je u sklopu nove opreme on zaseban element. Na osima su izabrani jači motori, pa je tako na osi X došlo do promjene motora snage 150 W za motor snage 200 W. Motor iste snage je stavljen na os Y, zamjenjujući tako motor snage 60 W, a na osi Z i R su postavljeni motori snage 100 W koji su zamijenili njihove motore snage 60 W na Z-osi, odnosno motor snage 30 W na R-osi. Aktuatori serije ISA/ISPA, odnosno njihove baze, linearne vodilice i kuglični vijci ostali su nepromijenjeni i nastavljaju se koristiti s novom upravljačkom opremom na kartezijskom koordinatnom robotu. Kako bi rekonstrukcija kartezijskog koordinatnog robota bila potpuna potrebno je bilo i isprogramirati robota kako bi se postigla mogućnost upravljanja njime i kasnije onda automatizacija njegovog rada. Programiranje robota je izvršeno u dva dijela, programski kod postavljen unutar PLC-a i python skripta pomoću koje smo dobili i grafičko sučelje za upravljanje robotom.

Osim same rekonstrukcije kartezijskog robota, kroz rad se može vidjeti i prednost ovakve vrste robota u odnosu na druge. Mogućnost relativno jednostavne promjene i izmjene elemenata i izgleda robota. Krajnji korisnici mogu promijeniti aktuatore, motore, upravljačke jedinice kartezijskog robota te ga mogu relativno jednostavno reprogramirati i prilagoditi korisničko sučelje kako bi zadovoljili promjenjive potrebe primjene u industriji.

Za kraj, ovim putem zahvalio bih se firmi Proton EL d.o.o. zbog njihove donacije upravljačke opreme, firme Delta electronics, laboratoriju za projektiranje izradbenih i montažnih sustava, čime su nam omogućili da izvršimo cjelovitu rekonstrukciju kartezijskog koordinatnog robota.

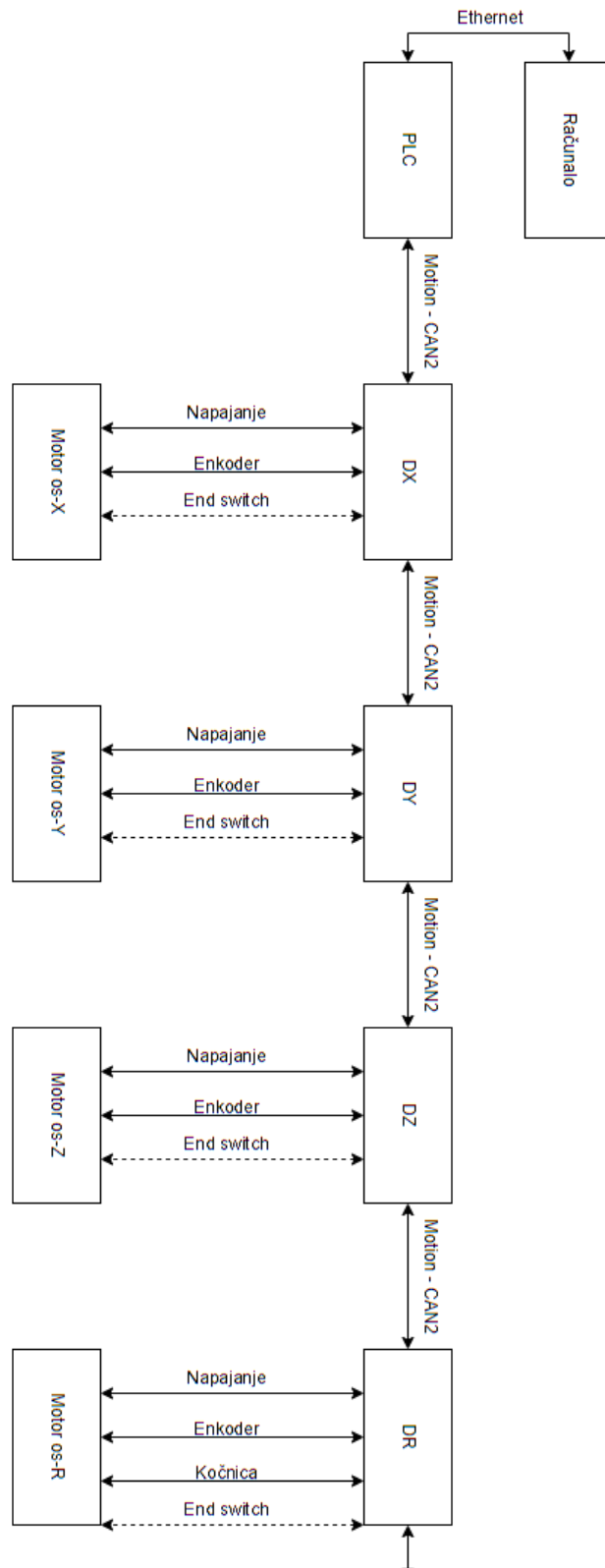
## Literatura

- [1] "The Top Six Types of Industrial Robots in 2020". DIY-Robotics. Pogledano 15.11.2022.
- [2] "OSHA Technical Manual (OTM) | Section IV: Chapter 4 - Industrial Robots and Robot System Safety | Occupational Safety and Health Administration". www.osha.gov. Pogledano 15.11.2022.
- [3] "Robots and robotic devices — Vocabulary". www.iso.org. 2012. Pogledano 15.11.2022.
- [4] "International Federation of Robotics". IFR International Federation of Robotics. Pogledano 15.11.2022.
- [5] "History of Industrial Robots" (PDF) Pogledano 15.11.2022.
- [6] Poole, Harry H. (2012). Fundamentals of Robotics Engineering
- [7] Z. Pandilov, V. Dukovski, Comparison of the characteristics between serial and parallel robots
- [8] Dabro A. – Prilagodba robota za glodaći ispitni postav, Diplomski rad 2016.
- [9] Delta electronics Inc. - Delta Programmable Logic Controller DVP Series 2010.
- [10] Wang Hesheng - Introduction to Robotics
- [11] Delta electronics Inc. - Delta AC Servo Drive & Motor ASDA-A2 Series
- [12] Delta electronics Inc. - DVP-15MC Series Motion Controller Operation Manual 2020.
- [13] International Automation Industry – Controller 2020.
- [14] International Automation Industry – Large-capacity Controller X-sel P/Q/QX
- [15] International Automation Industry – Linear axis system
- [16] International Automation Industry - X-SEL Controller J/K/KE Type, Operational manual 2015.

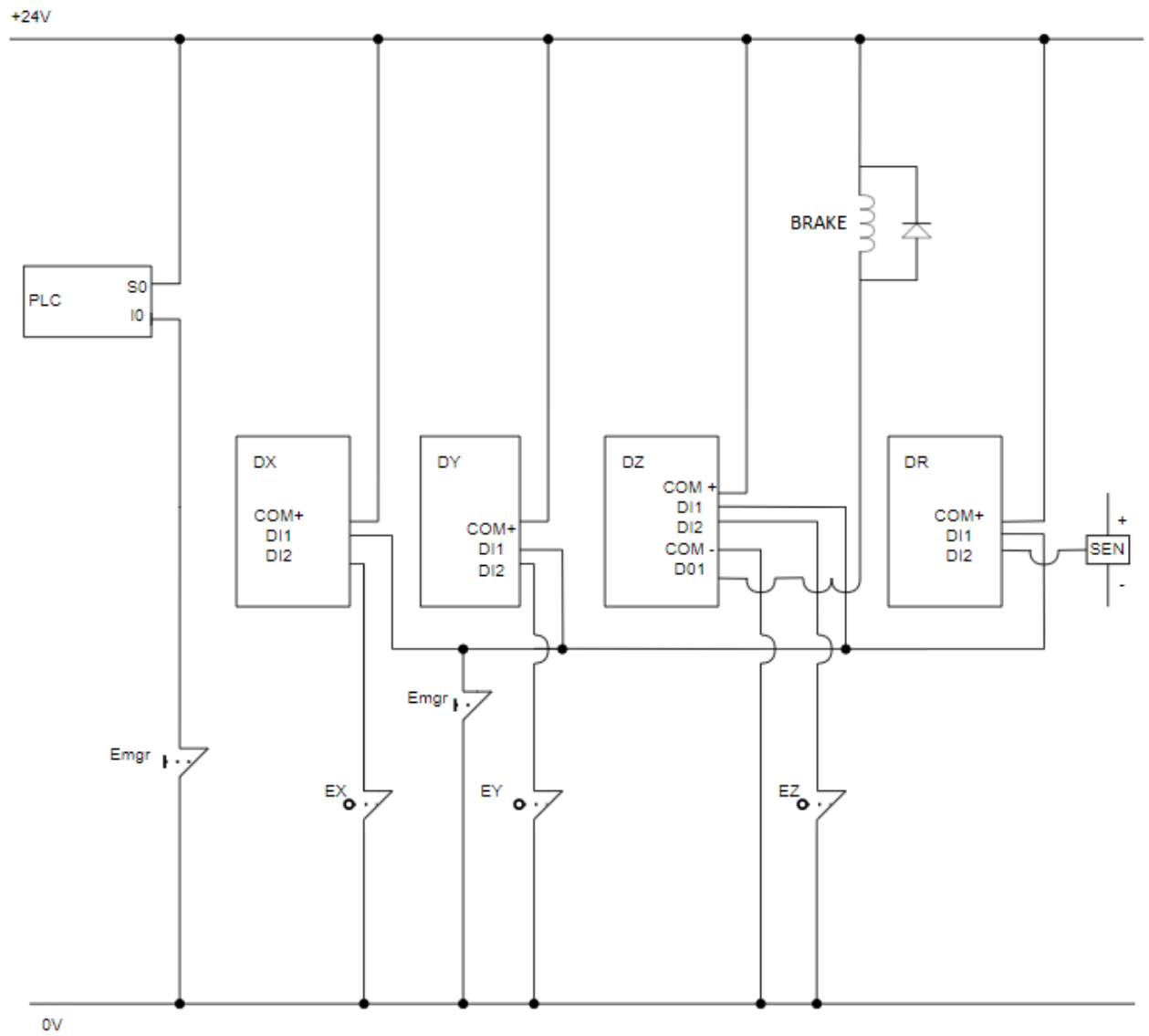
## **Prilozi**

- I. Blok shema spoja PLC-a, servo pogona i servo motora
- II. Shema spajanja kontakata PLC-a i servo pogona s ostalim komponentama
- III. Programski kod PLC-a (Ladder diagram)
- IV. Popis signala
- V. Python skripta za čitanje G-koda
- VI. Tehnička dokumentacija

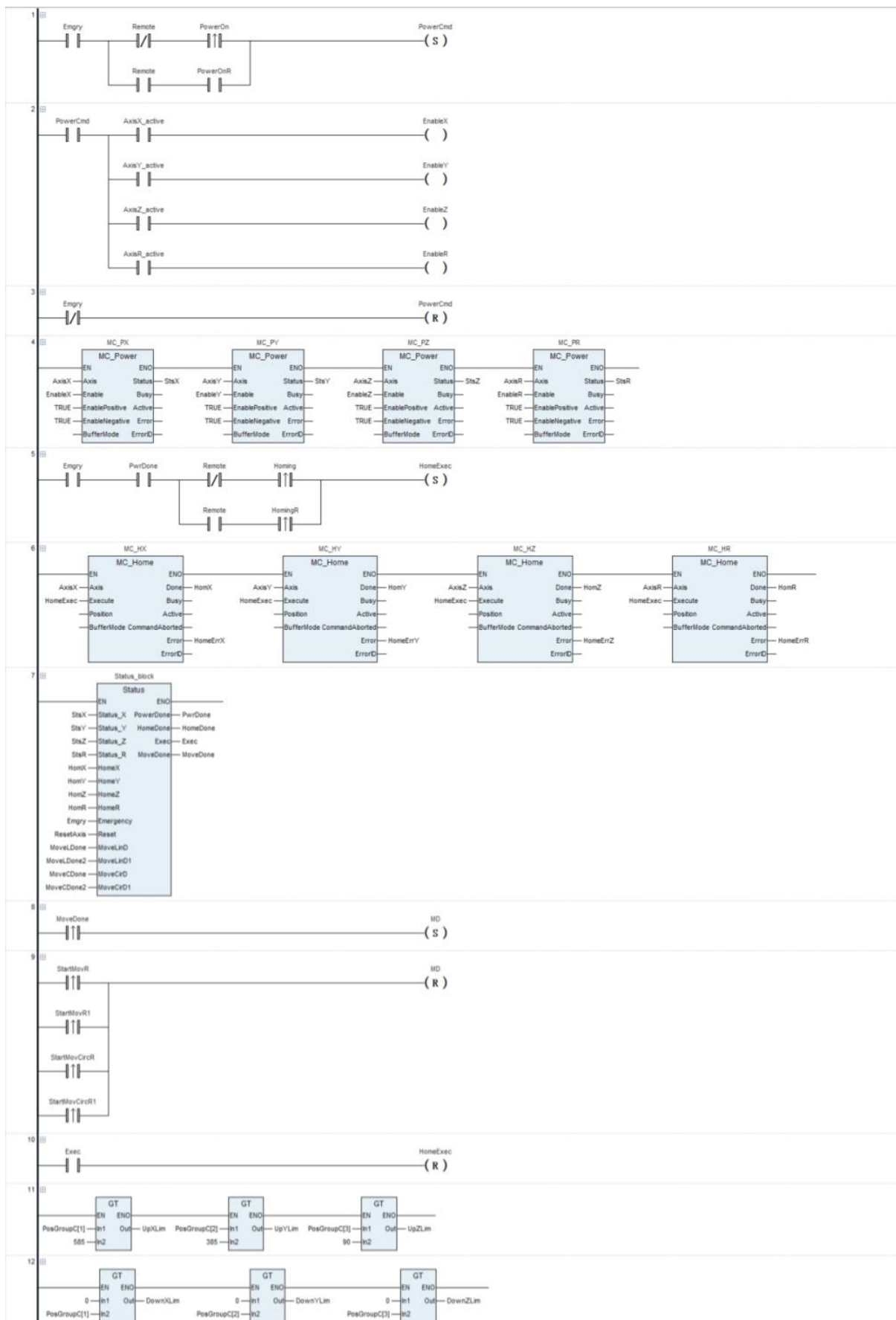
I. Blok shema spoja PLC-a, servo pogona i servo motora

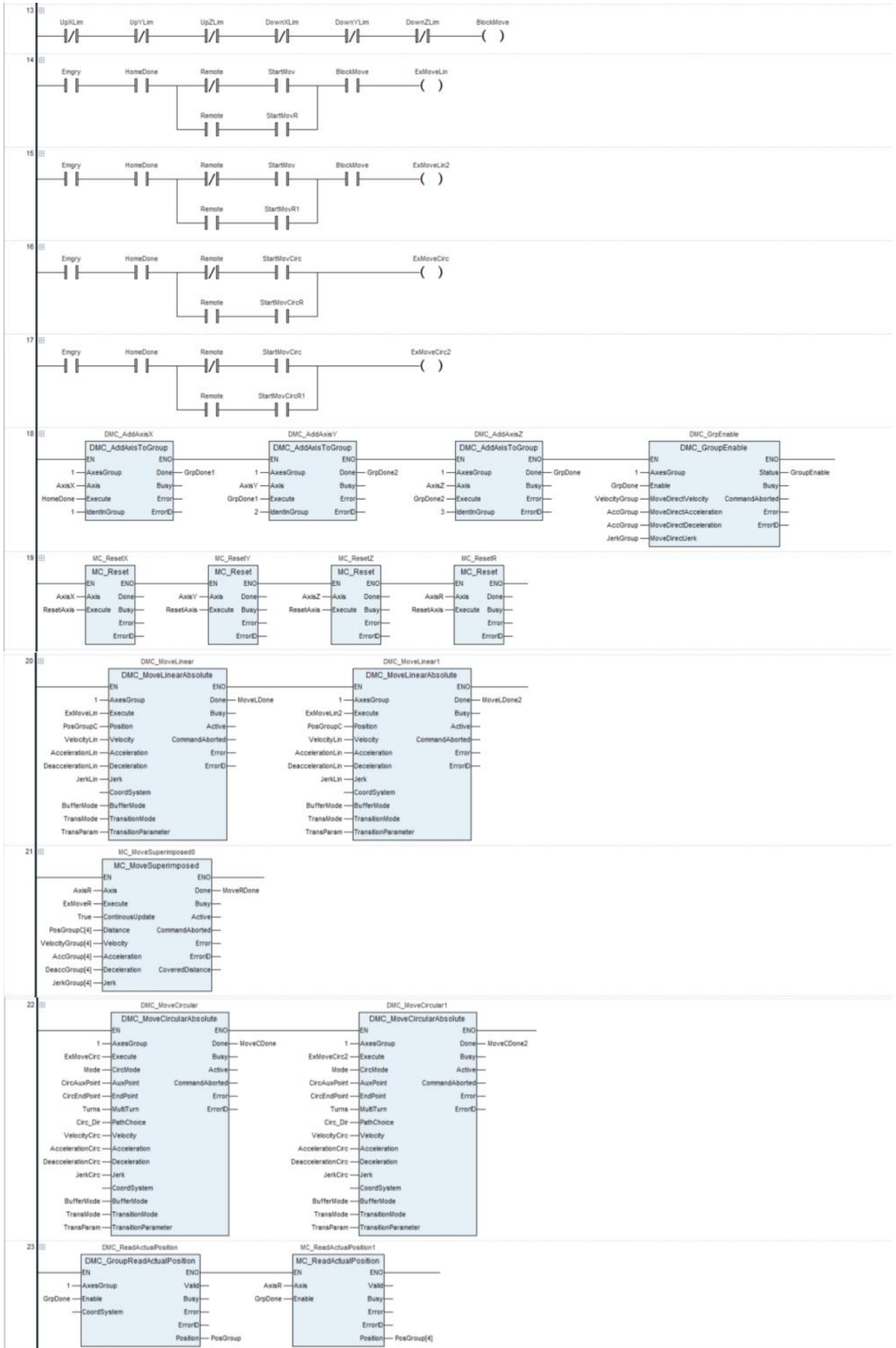


## II. Shema spajanja kontakata PLC-a i servo pogona s ostalim komponentama



### III. Programski kod PLC-a (Ladder diagram)





## IV. Popis signala

Index	Scope	Name	Address	Data Type	Initial Value	Comment
1	VAR	Emgry	%IX0.0	BOOL		Emergency button
2	VAR	Remote	%MX6.0	BOOL	TRUE	Remote switch
3	VAR	PowerOn	%IX0.1	BOOL		Power ON button
4	VAR	PowerOnR	%MX0.1	BOOL	FALSE	Power ON button
5	VAR	Homing	%IX0.2	BOOL		Homing button
6	VAR	HomingR	%MX0.2	BOOL	FALSE	Remote homing
7	VAR	StartMov	%IX0.3	BOOL		Start move button
8	VAR	StartMovR	%MX0.3	BOOL	FALSE	Start move remote
9	VAR	ResetAxis	%MX0.4	BOOL		Reset Error state on axis
10	VAR	StartMovCirc	%IX0.5	BOOL		Start circular move button
11	VAR	StartMovCircR	%MX0.5	BOOL	FALSE	Start circular move remote
12	VAR	ExMoveLin	%MX4.0	BOOL	FALSE	Execute linear movement
13	VAR	ExMoveLin2		BOOL	FALSE	Execute linear movement 2
14	VAR	ExMoveR	%MX6.2	BOOL	FALSE	Execute R axis movement
15	VAR	ExMoveCirc	%MX4.1	BOOL	FALSE	Execute circular movement
16	VAR	ExMoveCirc2		BOOL	FALSE	Execute circular movement 2
17	VAR	MoveDone		BOOL	FALSE	Linear move done
18	VAR	MD	%MX4.2	BOOL	FALSE	Linear move done 2
19	VAR	MoverDone	%MX5.7	BOOL	FALSE	R axis movement done
20	VAR	MoveCDone		BOOL	FALSE	Circular move done
21	VAR	MoveCDone2		BOOL	FALSE	Circular move done 2
22	VAR	AxisX	%MB40	USINT	1	X axis ID
23	VAR	AxisY	%MB41	USINT	2	Y axis ID
24	VAR	AxisZ	%MB42	USINT	3	Z axis ID
25	VAR	AxisR	%MB43	USINT	4	R axis ID
26	VAR	MC_PX		MC_Power		Starting axis X block
27	VAR	MC_PY		MC_Power		Starting axis Y block
28	VAR	MC_PZ		MC_Power		Starting axis Z block
29	VAR	MC_PR		MC_Power		Starting axis R block
30	VAR	MC_HX		MC_Home		Homing X block
31	VAR	MC_HY		MC_Home		Homing Y block
32	VAR	MC_HZ		MC_Home		Homing Z block
33	VAR	MC_HR		MC_Home		Homing R block
34	VAR	DMC_AddAxisX		DMC_AddAxisToGroup		Add axis X block
35	VAR	DMC_AddAxisY		DMC_AddAxisToGroup		Add axis Y block
36	VAR	DMC_AddAxisZ		DMC_AddAxisToGroup		Add axis Z block
37	VAR	DMC_GrpEnable		DMC_GroupEnable		Enable group block
38	VAR	DMC_MoveLinear		DMC_MoveLinearAbsolute		Move linear block
39	VAR	DMC_MoveCircular		DMC_MoveCircularAbsolute		Move circular block
40	VAR	DMC_ReadActualPosition		DMC_GroupReadActualPosition		Read position block
41	VAR	VelocityLin	%ML38	LREAL	50	Velocity for linear movement
42	VAR	AccelerationLin	%ML46	LREAL	100	Acc for linear movement
43	VAR	DeaccelerationLin	%ML54	LREAL	100	Deacc for linear movement
44	VAR	JerkLin	%ML62	LREAL	200	Jerk for linear movement
45	VAR	VelocityCirc	%ML39	LREAL	50	Velocity for circular movement
46	VAR	AccelerationCirc	%ML47	LREAL	100	Acc for circular movement
47	VAR	DeaccelerationCirc	%ML55	LREAL	100	Deacc for circular movement
48	VAR	JerkCirc	%ML63	LREAL	200	Jerk for circular movement
49	VAR	VelocityGroup	%ML34	ARRAY[1..8] OF LREAL	[50, 50, 50, 20]	Velocity for axis [X,Y,Z,R,Lin,Circ]
50	VAR	AccGroup	%ML42	ARRAY[1..8] OF LREAL	[100, 100, 100, 40]	Acceleration for axis [X,Y,Z,R,Lin,Circ]
51	VAR	DeaccGroup	%ML50	ARRAY[1..8] OF LREAL	[100, 100, 100, 40]	Deceleration for axis [X,Y,Z,R,Lin,Circ]; i
52	VAR	JerkGroup	%ML58	ARRAY[1..8] OF LREAL	[200, 200, 200, 80]	Jerk for axis [X,Y,Z,R,Lin,Circ]
53	VAR	PosGroupC	%ML8	ARRAY[1..8] OF LREAL		Actual position for axis [X,Y,Z]
54	VAR	PosGroup	%ML16	ARRAY[1..8] OF LREAL		Position command for axis [X,Y,Z]
55	VAR	CircAuxPoint	%ML32	ARRAY [1..2] OF LREAL		Circle center[I,J] or radius[R,-]
56	VAR	CircEndPoint	%ML24	ARRAY [1..8] OF LREAL		Circle endpoint [X,Y,Z]
57	VAR	Radius	%ML32	LREAL		Use this for radius instead array
58	VAR	PowerCmd	%MX1.0	BOOL	FALSE	Power ON command
59	VAR	EnableX	%MX1.1	BOOL	FALSE	Enable X axis
60	VAR	EnableY	%MX1.2	BOOL	FALSE	
61	VAR	EnableZ	%MX1.3	BOOL	FALSE	Enable Z axis
62	VAR	EnableR	%MX1.4	BOOL	FALSE	Enable R axis
63	VAR	HomeErrX	%MX3.4	BOOL	FALSE	Homing X err
64	VAR	HomeErrY	%MX3.5	BOOL	FALSE	Homing Y err
65	VAR	HomeErrZ	%MX3.6	BOOL	FALSE	Homing Z err
66	VAR	HomeErrR	%MX3.7	BOOL	FALSE	Homing R err
67	VAR	GrpDone	%MX4.4	BOOL	FALSE	Group created
68	VAR	GrpDone1	%MX4.5	BOOL	FALSE	Added first axis to group
69	VAR	GrpDone2	%MX4.6	BOOL	FALSE	Added second axis to group
70	VAR	GroupEnable	%MX4.7	BOOL	FALSE	Enabled group
71	VAR	Debug_variables		BOOL		Break line for debug
72	VAR	AxisX_active	%MX5.0	BOOL	TRUE	Is axis X allowed
73	VAR	AxisY_active	%MX5.1	BOOL	TRUE	Is axis Y allowed
74	VAR	AxisZ_active	%MX5.2	BOOL	TRUE	Is axis Z allowed
75	VAR	AxisR_active	%MX5.3	BOOL	TRUE	Is axis R allowed
76	VAR	HomeExec		BOOL		
77	VAR	MC_ResetX		MC_Reset		
78	VAR	MC_ResetY		MC_Reset		
79	VAR	MC_ResetZ		MC_Reset		
80	VAR	MC_ResetR		MC_Reset		
81	VAR	OutOfBoundsX		BOOL		
82	VAR	OutOfBoundsY		BOOL		
83	VAR	OutOfBoundsZ		BOOL		
84	VAR	UpXLim		BOOL		
85	VAR	UpYLim		BOOL		
86	VAR	UpZLim		BOOL		
87	VAR	DownXLim		BOOL		
88	VAR	DownYLim		BOOL		
89	VAR	DownZLim		BOOL		
90	VAR	BlockMove	%MX0.6	BOOL		
91	VAR	Direction		MC_Direction	2	
92	VAR	MC_MoveAbsolute0		MC_MoveAbsolute		
93	VAR	MC_ReadActualPosition0		MC_ReadActualPosition		
94	VAR	MoveLPAxis		BOOL		
95	VAR	MoveLRAxis		BOOL		
96	VAR	Status_block		Status		
97	VAR	StsX		BOOL		
98	VAR	StsY		BOOL		
99	VAR	StsZ		BOOL		
100	VAR	StsR		BOOL		



101	VAR	☐☐ PwrDone		BOOL		
102	VAR	☐☐ HomeDone		BOOL		
103	VAR	☐☐ HomX		BOOL		
104	VAR	☐☐ HomY		BOOL		
105	VAR	☐☐ HomZ		BOOL		
106	VAR	☐☐ HomR		BOOL		
107	VAR	☐☐ TEST		BOOL		
108	VAR	☐☐ Exec		BOOL		
109	VAR	☐☐ DMC_AddAxisR		DMC_AddAxisToGroup		
110	VAR	☐☐ GrpDone3		BOOL		
111	VAR	☐☐ MC_MoveSuperimposed0		MC_MoveSuperimposed		
112	VAR	☐☐ MC_ReadActualPosition1		MC_ReadActualPosition		
113	VAR	☐☐ Turns	%MW7	UINT	0	
114	VAR	☐☐ Circ_Dir	%MW6	INT	0	
115	VAR	☐☐ TransMode	%MW5	MC_Transition_Mode	3	
116	VAR	☐☐ TransParam	%ML2	LREAL	2	
117	VAR	☐☐ BufferMode		MC_Buffer_Mode	3	
118	VAR	☐☐ Mode	%MW4	INT	0	
119	VAR	☐☐ DMC_MoveLinear1		DMC_MoveLinearAbsolute		
120	VAR	☐☐ DMC_MoveCircular1		DMC_MoveCircularAbsolute		
121	VAR	☐☐ MoveLDone		BOOL		
122	VAR	☐☐ MoveLDone2		BOOL		
123	VAR	☐☐ StartMovR1	%MX5.4	BOOL		
124	VAR	☐☐ StartMovCircR1	%MX5.5	BOOL		
125	VAR	☐☐ ExMoveLin1		BOOL		
126	VAR	☐☐ ExMoveCirc1		BOOL		

## V. Python skripta za čitanje G-koda

```
class GCodeParser:
    def __init__(self, fp: str):
        self.param_codes = ['X', 'Y', 'Z', 'R', 'F', 'I', 'J', 'K', 'S', 'P', 'D', 'H']
        self.lines = []
        self.parsed = {}
        self.params = {}
        self.data = {}
        self.circ_plane = 0
        self.block_array = []
        with open(fp, 'r') as file:
            for line in file.readlines():
                self.lines.append(line.strip('\n'))
        self.current_line = 0

    def linear_move_command(self, move_type: int):
        self.parsed['PARAMS'] = self.data
        self.parsed['MOVE_TYPE'] = move_type

    def circular_move_command(self, move_type: int):
        self.parsed['PARAMS'] = self.data
        self.parsed['MOVE_TYPE'] = move_type
        self.parsed['PLANE'] = self.circ_plane
        self.parsed['DIRECTION'] = 0 if move_type == 2 else 1

    def special_commands(self, command: str):
        self.parsed['COMMAND'] = command

    def machine_command(self, command: str, param: str = None):
        self.parsed['M_COMMAND'] = command
        if param is not None:
            self.parsed['COMMAND_PARAM'] = param
        pass

    def process_lines(self, line_num: int):
        line = self.lines[line_num]
        proc_line = line.split(' ')
        for idx, comm in enumerate(proc_line):
            if comm.startswith('G'):
                base_command = int(comm[1:])
                if base_command == 0 or base_command == 1:
                    self.params = proc_line[idx:]
                    self.parse_params()
                    self.linear_move_command(base_command)
                elif base_command == 2 or base_command == 3:
                    self.params = proc_line[idx:]
                    self.parse_params()
                    self.circular_move_command(base_command)
                elif base_command == 4:
                    self.special_commands('DWELL')
                elif base_command == 17:
                    self.circ_plane = 'xy'
                elif base_command == 18:
                    self.circ_plane = 'xz'
                elif base_command == 19:
                    self.circ_plane = 'yz'
                elif base_command == 28:
                    self.special_commands('HOME')
```

```

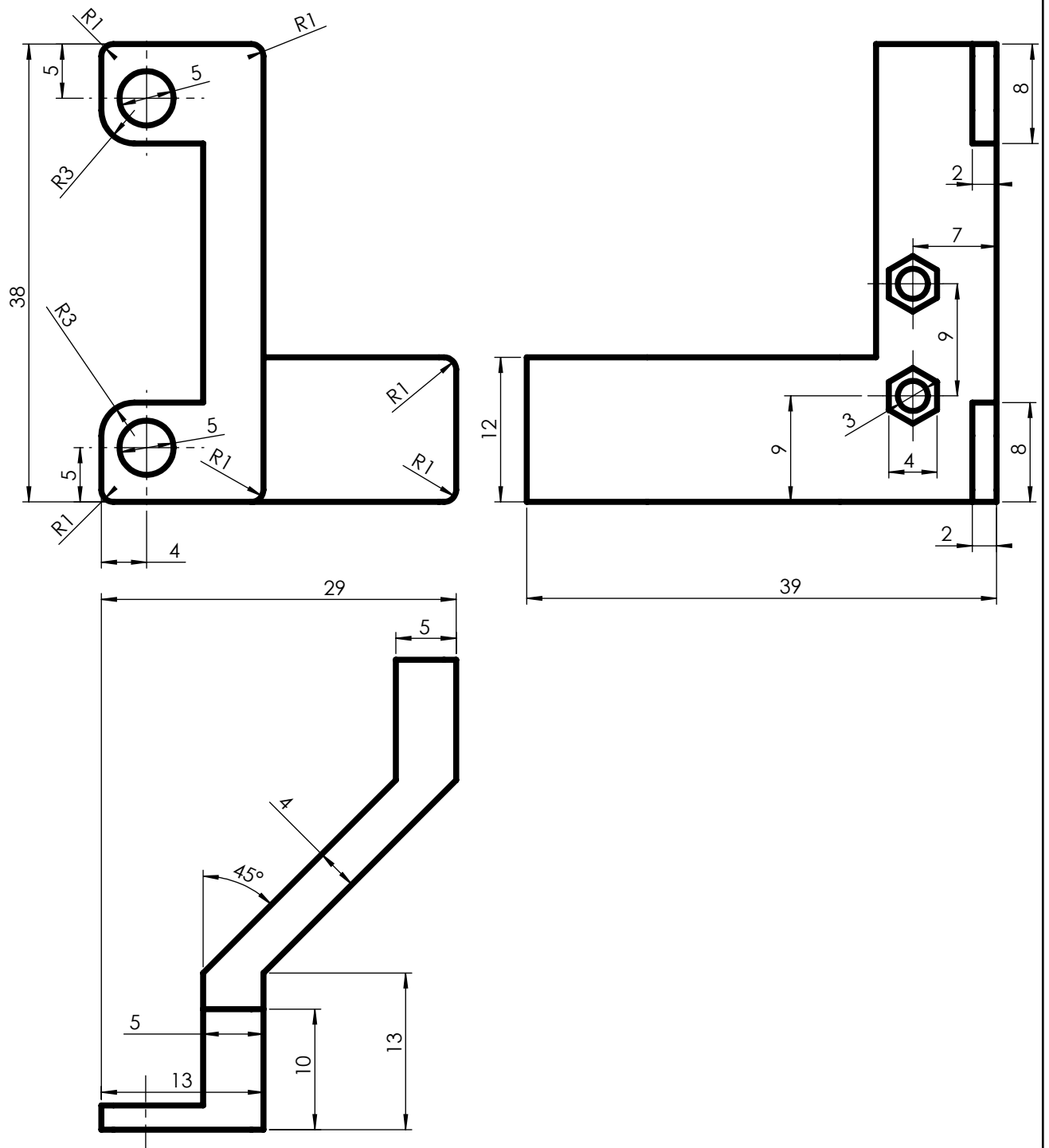
elif base_command == 90:
    self.special_commands('ABSOLUTE')
elif base_command == 91:
    self.special_commands('RELATIVE')
else:
    print('COMMAND NOT IMPLEMENTED')
if comm.startswith('M'):
    base_command = int(comm[1:])
    if base_command == 0:
        self.machine_command('PAUSE')
    if base_command == 2:
        self.machine_command('STOP_WO_RESTART')
    if base_command == 30:
        self.machine_command('STOP')
    if base_command == 62:
        address = proc_line[idx+1][1:]
        self.machine_command('BIT ON', address)
    if base_command == 63:
        address = proc_line[idx+1][1:]
        self.machine_command('BIT OFF', address)
if comm.startswith('N'):
    line_number = int(comm[1:])
    self.parsed['LINE'] = line_number

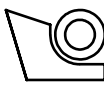
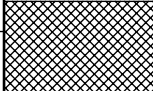
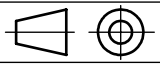
def parse_params(self):
    for param in self.params:
        if param.startswith(tuple(self.param_codes)):
            name = param[0]
            val = float(param[1:])
            self.data[name] = val

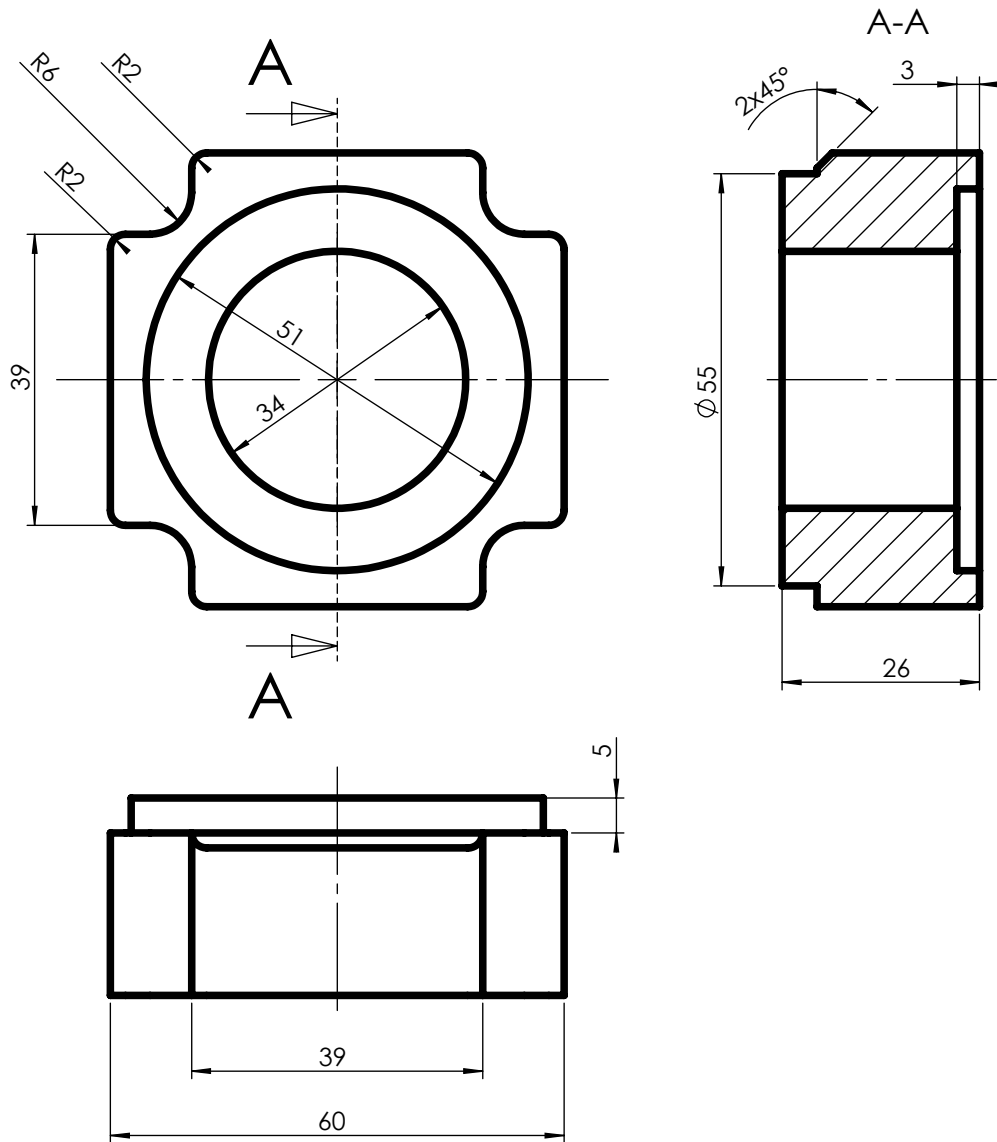
def next_command(self):
    if self.current_line >= len(self.lines):
        return 'EOF'
    self.parsed = {}
    self.process_lines(self.current_line)
    self.current_line += 1
    return self.parsed

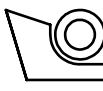
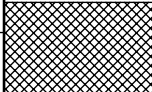
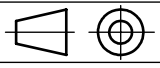
def block_chooser(self, move):
    try:
        last_block = self.block_array[-1]
    except IndexError:
        last_block = None
    if move in (0, 1):
        if last_block is None:
            self.block_array.append(0)
        elif last_block == 0:
            self.block_array.append(1)
        elif last_block in (1, 2, 3):
            self.block_array.append(0)
    if move in (2, 3):
        if last_block is None:
            self.block_array.append(2)
        elif last_block == 2:
            self.block_array.append(3)
        elif last_block in (0, 1, 3):
            self.block_array.append(2)
    print(self.block_array)
    return self.block_array[-1]

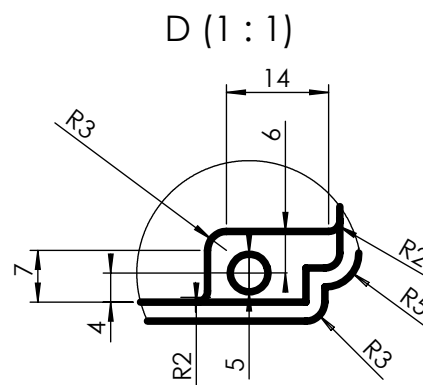
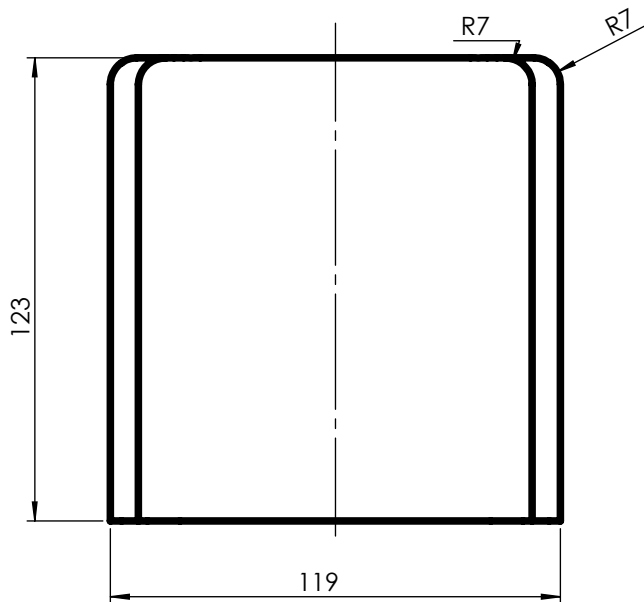
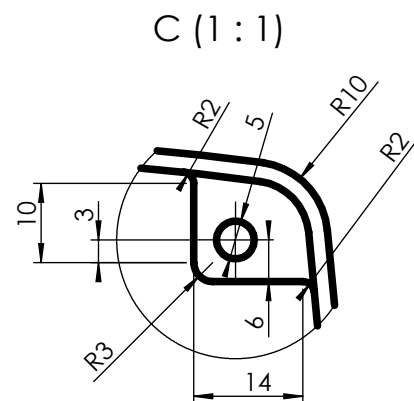
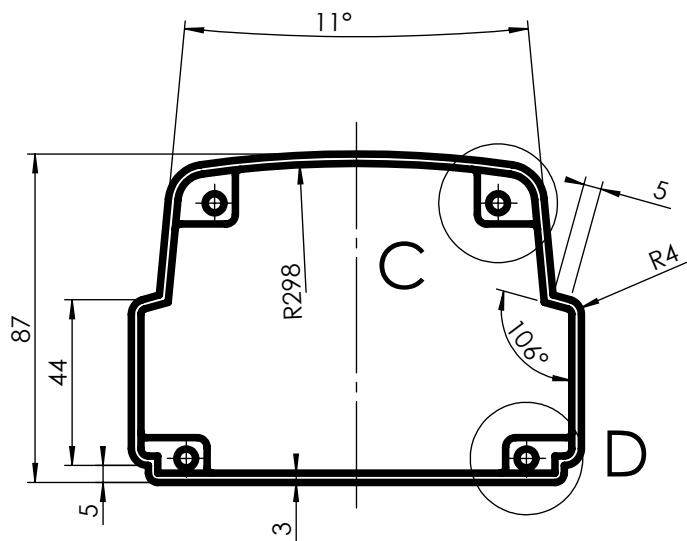
```


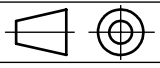


	Datum	Ime i prezime	Potpis	 <b>FSB Zagreb</b>
Projektirao		Krešimir Major		
Razradio				
Crtao		Krešimir Major		
Pregledao				
Objekt:			Objekt broj:	
			R. N. broj:	
Napomena:				Kopija
				
Materijal:	PLA	Masa:		
	Naziv:		Pozicija:	Format: A4
Mjerilo originala	Držač krajnjeg prekidača			Listova: 1
M 2:1	Crtež broj: Poz - 1			List: 1



	Datum	Ime i prezime	Potpis	 <b>FSB Zagreb</b>
Projektirao		Krešimir Major		
Razradio				
Crtao		Krešimir Major		
Pregledao				
Objekt:			Objekt broj:	
			R. N. broj:	
Napomena:				Kopija
				
Materijal:	PLA	Masa:		
	Naziv:		Pozicija:	Format: A4
Mjerilo originala	Distancer X-osi			Listova: 1
M1:1	Crtež broj: Poz - 2			List: 1



	Datum	Ime i prezime	Potpis	
Projektirao		Krešimir Major		
Razradio				
Crtao		Krešimir Major		
Pregledao				
Objekt:			Objekt broj:	<div style="background-color: #cccccc; width: 100%; height: 100%;"></div>
			R. N. broj:	
Napomena:				Kopija
Materijal:	PLA	Masa:		<div style="background-color: #cccccc; width: 100%; height: 100%;"></div>
 Mjerilo originala	Naziv:		Pozicija:	
M1:2	<b>Kućište X-osi</b>		Format: A4	
Crtež broj: Poz - 3			Listova: 1	<div style="background-color: #cccccc; width: 100%; height: 100%;"></div>
			List: 1	