

# Implementacija pozicioniranja pokretne trake

---

Lukavečki, Vjeko

Master's thesis / Diplomski rad

2024

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:057609>

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

Download date / Datum preuzimanja: **2025-03-11**



Repository / Repozitorij:

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



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**IMPLEMENTACIJA POZICIONIRANJA POKRETNE TRAKE**

Rijeka, rujan 2024.

Vjeko Lukavečki

0069086897

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**IMPLEMENTACIJA POZICIONIRANJA POKRETNE TRAKE**

Mentor: prof. dr. sc. Dario Matika

Komentor: v. asist. dr. sc. Nikola Turk

Rijeka, rujan 2024.

Vjeko Lukavečki

0069086897

Rijeka, 14.03.2024.

Zavod:                   Zavod za automatiku i elektroniku  
Predmet:                Automatizacija postrojenja i procesa

## ZADATAK ZA DIPLOMSKI RAD

Pristupnik:           **Vjeko Lukavečki (0069086897)**  
Studij:                Sveučilišni diplomski studij elektrotehnike (1300)  
Modul:                 Automatika (1331)

Zadatak:              **Implementacija pozicioniranja pokretne trake / Conveyor belt positioning implementation**

### Opis zadatka:

Potrebno je projektirati i izraditi upravljački program za pozicioniranje predmeta na pokretnoj traci u otvorenoj petlji i zatvorenoj petlji. Izrađeni upravljački program potrebno je implementirati i testirati na stvarnom sustavu pokretne trake. Zadatak diplomskog rada uključuje detaljan opis sustava, razvoj programske aplikacije u programskom okruženju Siemens Tia Portal, implementaciju i funkcionalno ispitivanje izrađene aplikacije na stvarnom sustavu pokretne trake. Upravljački program potrebno je implementirati na PLC S7-1200 koji upravlja pozicijom pokretne trake posredstvom frekvencijskog pretvarača Yaskawa i asinkronog motora.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku:    20.03.2024.

Mentor:  
prof. dr. sc. Dario Matika

Komentor:  
dr. sc. Nikola Turk

Predsjednik povjerenstva za  
diplomski ispit:  
prof. dr. sc. Dubravko Franković

## IZJAVA

Sukladno Pravilniku o diplomskom radu, diplomskom ispitu i završetku diplomskih sveučilišnih studija Tehničkog fakulteta u Rijeci, izjavljujem da sam samostalno izradio diplomski rad prema zadatku koji mi je uručen 20.03.2024.

Rijeka, rujan 2024.



---

Vjeko Lukavečki

0069086897

## **Zahvala**

Zahvaljujem svom mentoru, dr. sc. Dariu Matiki, i komentoru, dr. sc. Nikoli Turku, na pruženoj pomoći, savjetima i uloženom vremenu prilikom pisanja diplomskog rada.

Također, zahvaljujem tvrtki Atinel d.o.o. na danoj mogućnosti izrade ovog rada u njihovoj firmi, te korištenja njihove opreme.

Na kraju, veliko hvala mojoj obitelji i prijateljima na podršci tokom svih ovih godina studiranja.

# SADRŽAJ

1. UVOD.....	3
2. UPRAVLJANJE I REGULACIJA SUSTAVA.....	4
2.1. Sustav u otvorenoj petlji.....	4
2.2. Sustav u zatvorenoj petlji.....	5
3. KOMPONENTE.....	6
3.1. Asinkroni motor.....	6
3.2. PLC.....	10
3.3. HMI.....	12
3.4. Frekvencijski pretvarač.....	13
3.5. Reduktor.....	18
3.6. Pokretna traka.....	19
4. OPIS SUSTAVA.....	20
4.1. Povezivanje komponenti .....	20
4.2. Opis rada sustava.....	22
5. TIA PORTAL.....	23
5.1. Ljestvičasti dijagram .....	24
6. PROGRAMSKA REALIZACIJA.....	26
6.1. Uspostava komunikacije uređaja.....	26
6.2. Programski kod.....	28
6.3. HMI vizualizacija.....	45
7. FUNKCIONALNO ISPITIVANJE SUSTAVA.....	52
8. ZAKLJUČAK.....	55
LITERATURA .....	56

SAŽETAK I KLJUČNE RIJEČI.....	58
DODATAK A: PROGRAMSKI KOD.....	60
POPIS SLIKA I TABLICA.....	67



## 1. UVOD

Automatizacija u industriji predstavlja ključnu komponentu moderne proizvodnje i poslovanja, omogućujući poduzećima da povećaju učinkovitost, smanje troškove i poboljšaju kvalitetu svojih proizvoda. Razvoj tehnologije je transformirao tradicionalne proizvodne procese, omogućujući brže i preciznije obavljanje zadataka. Automatizacija smanjuje potrebu za ručnim radom, minimizira ljudske pogreške, poboljšava sigurnost na radnom mjestu i omogućuje poduzećima da se lakše prilagode tržišnim uvjetima.

Pokretne trake predstavljaju jedan od najranijih i najvažnijih primjera automatizacije u industriji. Uvođenje pokretnih traka početkom 20. stoljeća postavilo je temelje za modernu industrijsku automatizaciju. Danas su pokretne trake neizostavan dio industrijskih sustava, integrirane s naprednim tehnologijama poput senzora, robota i softvera za upravljanje procesima.

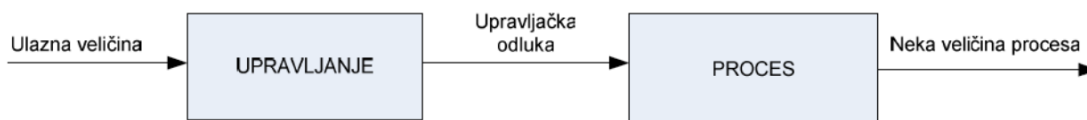
U sklopu ovog diplomskog rada izrađen je upravljački program za pozicioniranje predmeta na pokretnoj traci u otvorenoj i zatvorenoj petlji, te je izrađeni upravljački program implementiran na stvarni model pokretne trake. Programska aplikacija je izrađena u programskom okruženju Siemens TIA Portal, te implementirana na PLC S7-1200, koji preko frekvencijskog pretvarača Yaskawa J1000 i asinkronog motora upravlja pozicijom pokretne trake. U ovom je radu dan detaljan opis sustava, te opis same programske aplikacije.

## 2. UPRAVLJANJE I REGULACIJA SUSTAVA

Upravljački sustav je skup uređaja koji upravljaju, usmjeravaju i reguliraju ponašanje drugih uređaja ili sustava pomoću upravljačkih petlji. Upravljački sustav ima ključnu ulogu u automatizaciji procesa, te je sastavni dio regulacije sustava. Postoje dva glavna tipa upravljačkog sustava: sustav u otvorenoj petlji i sustav u zatvorenoj petlji.

### 2.1. Sustav u otvorenoj petlji

Upravljački sustav u otvorenoj petlji (engl. *Open-loop control system*) je vrsta sustava upravljanja u kojem je ulazna veličina u proces neovisna o izlaznoj veličini iz procesa. Drugim riječima, sustav ne koristi povratnu informaciju kako bi utvrdio je li njegov izlaz postigao željenu ulaznu vrijednost.



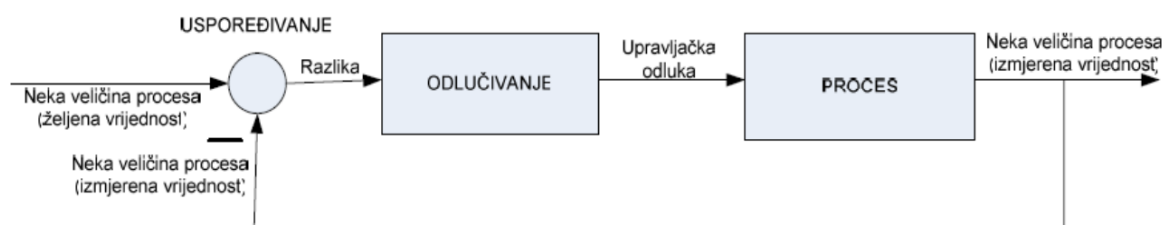
Slika 2.1. Shema upravljačkog sustava u otvorenoj petlji.[1]

Sustav u otvorenoj petlji se sastoji od ulazne veličine koja predstavlja željenu vrijednost koju sustav treba postići. Zatim, upravljački blok, to jest, regulator obrađuje ulazne vrijednosti te na temelju njih generira upravljački signal koji dovodimo na proces. Procesni blok predstavlja fizički sustav kojim upravljamo, dok izlazna veličina predstavlja odziv sustava na upravljački signal koji generira regulator.[2]

Budući da sustav ne dobiva povratne informacije o stvarnom trenutnom stanju izlazne veličine, sklon je odstupanjima od željene vrijednosti ako dođe do vanjske smetnje ili poremećaja unutar samog sustava. Ovaj problem rješavamo dodavanjem povratne veze (engl. *Feedback*) čime dobivamo sustav u zatvorenoj petlji.

## 2.2. Sustav u zatvorenoj petlji

Upravljački sustav u zatvorenoj petlji (engl. *Closed-loop control system*) je vrsta sustava za upravljanje gdje se izlazna veličina sustava vraća na ulaz. To se postiže pomoću povratne veze koja omogućava usporedbu izmjerene vrijednosti neke veličine reguliranog procesa sa njenom željenom vrijednosti, te na temelju razlike tih dviju veličina sustav odlučuje kako usmjeriti proces. Sustavi u zatvorenoj petlji su poželjni u sustavima upravljanja jer omogućuju automatsko održavanje željenog stanja nekog procesa, to jest, automatsku regulaciju.



Slika 2.2. Shema upravljačkog sustava u zatvorenoj petlji.[1]

Sustav u zatvorenoj petlji sadrži iste komponente kao i u otvorenoj petlji, uz dodatak povratne veze i zbrajala. Kao što je već spomenuto, povratna veza je dio izlaznog signala koji se dovodi u zbrajalo radi usporedbe s željenim referentnim ulazom. Zbrajalo zatim mjeri razliku između željenog referentnog ulaza i stvarnog izlaza sustava, te generira signal pogreške koji se dovodi na regulator.[3]

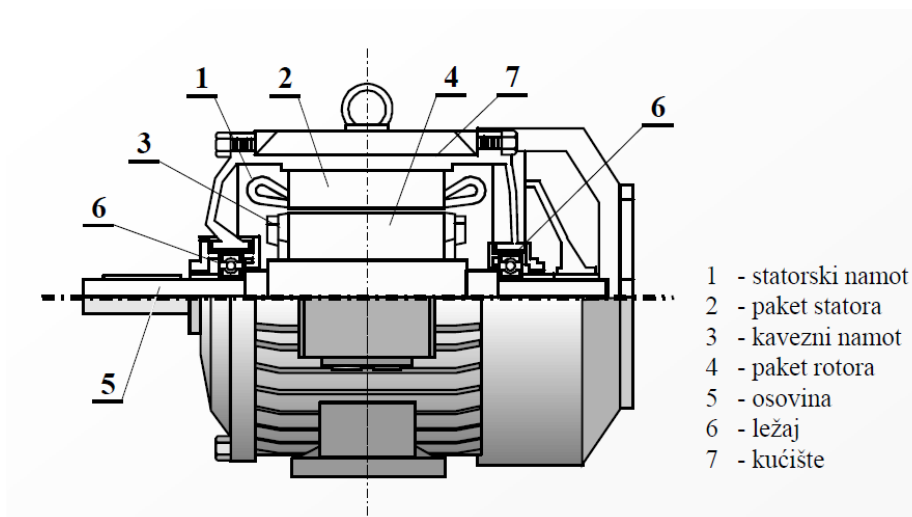
Potrebno je i naglasiti da regulator u zatvorenom sustavu kontinuirano ispravlja i prilagođuje upravljački signal na temelju povratne informacije, dok u otvorenom sustavu regulator generira upravljački signal samo na temelju unaprijed definiranih parametara, bez obzira na stvarni izlaz sustava. Ovo omogućuje zatvorenim sustavima veću preciznost, te bolju stabilnost sustava.[3]

### 3. KOMPONENTE

#### 3.1. Asinkroni motor

##### 3.1.1. Princip rada asinkronog motora

Asinkroni motor je izmjenični elektromotor u kojem rotirajuće magnetsko polje statora elektromagnetskom indukcijom inducira napon, odnosno struju na rotoru koja posljedično uzrokuje moment i gibanje. Rotor asinkronog motora može biti kaveznog tipa ili klizno kolutnog tipa, međutim, u svrhu ovog rada koristiti se trofazni kavezni asinkroni motor, čiju građu možemo vidjeti u nastavku na Slici 3.1.

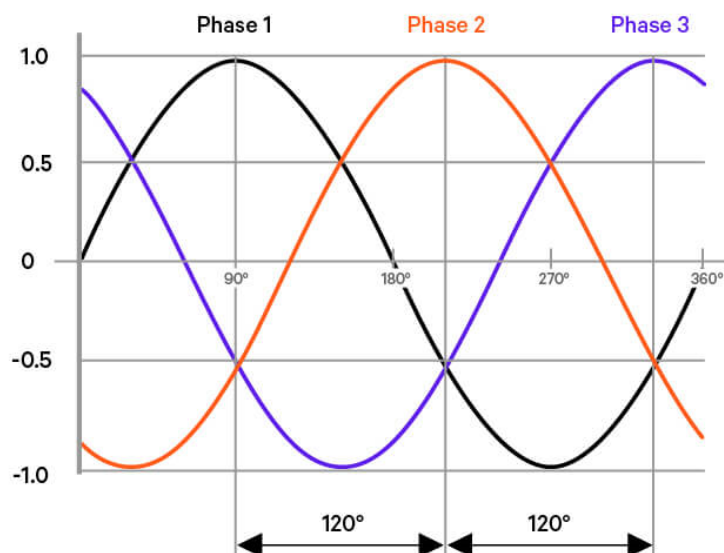


Slika 3.1. Kavezni asinkroni motor s označenim dijelovima.[4]

Trofazni asinkroni motor na statoru ima tri prostorno pomaknuta namota, odnosno faze, kroz koje teku tri struje fazno pomaknute za  $120^\circ$  (Slika 3.2.). Sva tri namota zajednički stvaraju magnetsko polje koje je uvijek istog iznosa ali neprestano mijenja smjer tako da dobijemo okretno magnetsko polje. Okretno magnetsko polje vrti se takozvanom sinkronom brzinom  $n_s$  po izrazu:

$$n_s = \frac{60 \cdot f}{p}, \quad (3.1)$$

gdje je:  $f$  frekvencija napajanja,  $p$  broj pari polova. Silnice okretnog magnetskog polja presijecaju vodiče rotora u kojima se, prema Faradayevom zakonu, inducira napon, te posljedično poteče struja. Budući da se vodiči kroz koje teče struja nalaze u magnetskom polju, na njih će djelovati Lorentzova sila koja će uzrokovati rotaciju rotora.[4]



Slika 3.2. Trofazna struja u statoru motora.

Rotor asinkronog motora ne može se okretati istom brzinom kao okretno magnetsko polje (sinkronom brzinom) jer bi tada rotorski vodiči i silnice okretnog magnetskog polja mirovali jedno u odnosu na drugo. Kao posljedica, ne bi došlo do presijecanja vodiča rotora od strane silnica okretnog magnetskog polja statora, pa ni inducirano napona, kao ni struje na rotoru, a prema tome ni sile. Relativna razlika između brzine rotora i brzine okretnog magnetskog polja naziva se klizanje  $s$ , te se matematički izračunava kao:

$$s = \frac{n_s - n}{n_s}, \quad (3.2)$$

gdje je:  $n_s$  sinkrona brzina,  $n$  brzina vrtnje rotora.[4]

U nastavku u Tablici 3.1. navedeni su podaci s natpisne pločice asinkronog motora korištenog u ovom radu.

*Tablica 3.1. Podaci asinkronog motora.*

Proizvođač	Končar MES
Broj motora ( <i>Code</i> )	A566485
Serijski broj i datum izrade ( <i>N</i> )	735580; 06/15
Broj faza ( $\sim$ )	3
Tipna oznaka ( <i>Type</i> )	5.5AZ63B-4; B14F75
Nazivni napon	D230/Y400 [V]
Nazivna frekvencija	50 [Hz]
Nazivna struja	1/0,6 [A]
Nazivna snaga	0,18 [kW]
Faktor snage ( $\cos\varphi$ )	0,72
Nazivna brzina vrtnje	1370 [okr/min]
Dozvoljena temperatura okoline ( <i>T<sub>a</sub></i> )	40 [°C]
Izolacijski sustav ( <i>Cl</i> )	F
Stupanj zaštite ( <i>IP</i> )	55
Vrsta pogona ( <i>S</i> )	S1

### 3.1.2. Skalarno upravljanje asinkronim motorom

Postoji više vrsta metoda upravljanja brzinom asinkronih motora, no budući da se u ovom radu koristi frekvencijski pretvarač, za kontrolu brzine koristit će se takozvana skalarna metoda, odnosno, U/f metoda. Skalarna metoda se temelji na osnovnoj jednadžbi za brzinu vrtnje asinkronog motora koja glasi:

$$n = \frac{60 \cdot f}{p} \cdot (1 - s), \quad (3.3)$$

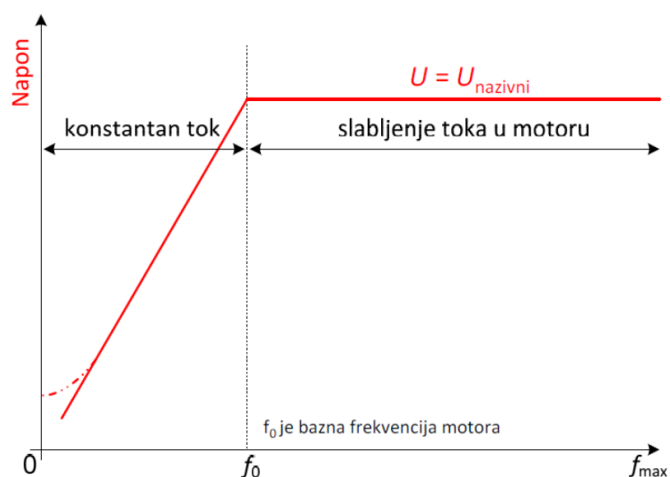
gdje je:  $n$  brzina vrtnje motora u okr/min,  $f$  frekvencija napajanja,  $p$  broj pari polova,  $s$  klizanje. Iz jednadžbe je vidljivo da je brzina vrtnje motora direktno proporcionalna frekvenciji, što upućuje na to da promjenom frekvencije možemo jednostavno upravljati brzinom vrtnje.

Međutim, frekvencija je motora također povezana s magnetskim tokom, što možemo vidjeti u jednadžbi za inducirani napon asinkronog motora:

$$U \approx E = 4,44 \cdot N \cdot f \cdot \Phi, \quad (3.4)$$

gdje je:  $U$  napon statora,  $E$  inducirani napon,  $N$  broj namotaja,  $f$  frekvencija,  $\Phi$  magnetski tok. Ako se promjeni frekvencija bez da se promjeni napon, doći će do promjene magnetskog toka, što može dovesti do podmagnetizacije ili zasićenja magnetskog kruga, ovisno o tome smanji li se ili poveća frekvencija pri konstantnom naponu. To će rezultirati smanjenjem efikasnosti i stabilnosti rada motora, zbog čega je potrebno magnetski tok održavati konstantnim.[5] To se postiže tako da se istovremeno mijenja napon i frekvencija motora prema izrazu:

$$\frac{U}{f} = konst. \quad (3.5)$$



Slika 3.3. Karakteristika skalarnog upravljanja.[5]

Kao što je već navedeno, upravljanje skalarnom metodom u praksi se postiže korištenjem frekvencijskog pretvarača, čiji je način rada opisan u potpoglavlju 3.4..

### 3.2. PLC

Programibilni logički kontroler (engl. *Programmable Logic Controller*) ili PLC je robusno industrijsko računalo prilagođeno za rad u automatiziranim proizvodnim procesima koji zahtijevaju visoku pouzdanost, jednostavno programiranje i dijagnostiku grešaka u procesu. Budući da je PLC namijenjen za rad u industrijskim uvjetima rada, potrebno je da bude otporan na vanjske smetnje poput električnog šuma, vibracija i udaraca, te da može raditi pri visokim i niskim temperaturama.

PLC se tipično sastoji od šest glavnih dijelova: centralna procesorska jedinica (CPU), memorija (RAM/ROM), ulazno/izlazni moduli, napajanje, programska jedinica, te komunikacijski modul.[6] CPU je odgovoran za izvršavanje kontrolnog programa pohranjenog u memoriji, te obavlja zadatke poput obrade podataka, donošenja odluka i komunikacije s drugim uređajima. RAM memorija se koristi za privremeno pohranjivanje podataka i varijabli tijekom izvođenja programa, dok se ROM memorija koristi za pohranu operativnog sustava PLC-a i korisničkog programa. Ulazni moduli primaju ulazne signale od senzora i šalju ih u CPU, te izlazni moduli šalju signale uređajima kojima upravljamo. Napajanje osigurava stabilno opskrbljivanje svih komponenti PLC-a električnom energijom. Programska jedinica je uređaj ili softver za pisanje,



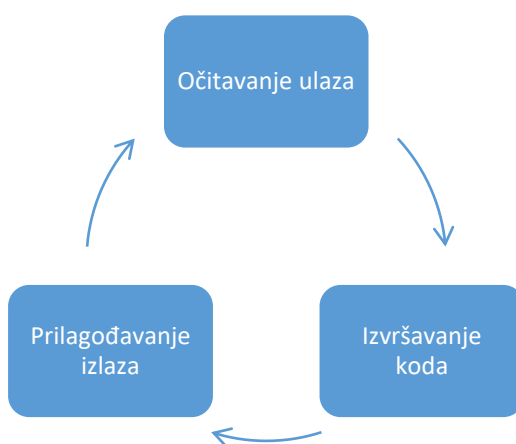
testiranje i izvršavanje programskog koda PLC-a. Komunikacijski moduli omogućuju PLC-u povezivanje s drugim PLC-ovima, računalima ili mrežama za razmjenu podataka.[6][7]

Na Slici 3.4. je prikazan PLC S7-1200 koji je korišten u ovom radu.



*Slika 3.4. SIMATIC S7-1200 CPU 1214C DC/DC/DC modul.[8]*

Princip rada PLC-a je takav da ulazni modul prima signal od senzora i drugih uređaja te ga prosljeđuje glavnom dijelu PLC-a. Glavni dio tada šalje signal CPU-u, koji zatim započinje izvršavanje programa u kompilatoru, koji prevodi programski kod iz izvornog programskog jezika u ciljani jezik. Nakon toga, program stvara upravljački signal koji se prosljeđuje u izlazni modul. Izlazni modul prima upravljački signal i šalje ga vanjskim uređajima kojim želimo upravljati. Ovaj proces rada se izvršava ciklički, to jest, PLC uzastopno izvršava program.[6][7]



*Slika 3.5. Ciklus PLC-a.*

Kao što je prije navedeno, pisanje koda za PLC vrši se na vanjskom uređaju, poput računala, u za to određenom softveru. U ovom radu korišten je razvojni program TIA Portal (Totally Integrated Automation Portal) koji je razvila kompanija Siemens. Za programiranje PLC-a postoji pet standardnih programskih jezika: Function Block Diagram (FBD), Sequential Function Charts (SFC), Structured Text (ST), Instruction List (IL), te Ladder Diagram (LD) koji je korišten u svrhu ovog rada.[6]

### 3.3. HMI

HMI (engl. *Human Machine Interface*) je uređaj koji omogućava interakciju između čovjeka i stroja. HMI korisniku pokazuje informacije u realnom vremenu poput stanja strojeva, podataka o proizvodnji i alarme, te daje korisniku mogućnost upravljanja samim procesom, npr. podešavanje postavki, promjenu parametara i slično. HMI se sastoji od zaslona na dodir (engl. *Touchscreen*) i HMI korisničkog sučelja preko kojeg korisnik komunicira sa strojem, odnosno, proizvodnim procesom.



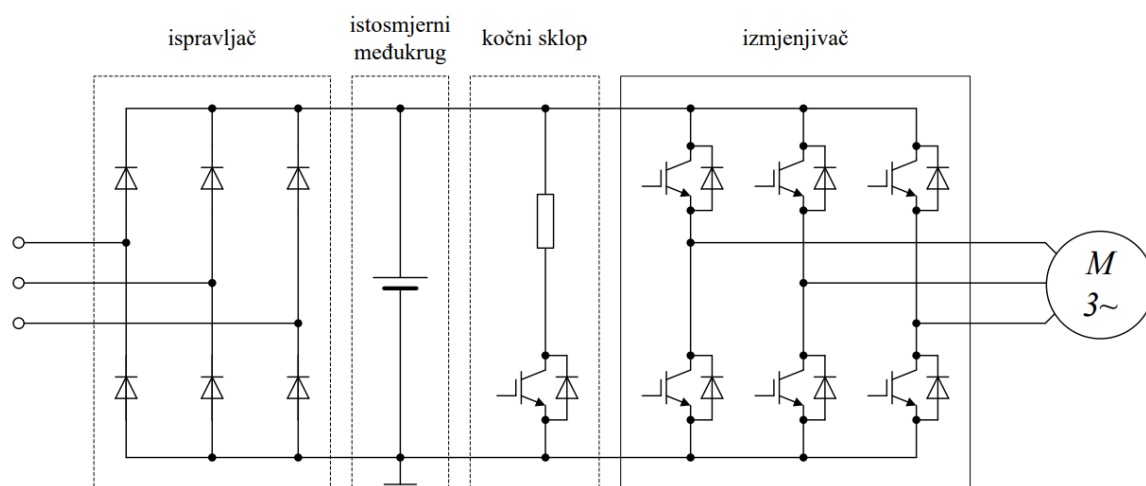
*Slika 3.6. HMI panel.*

U ovom je radu korištena simulacija HMI-a unutar programa TIA Portal.

### 3.4. Frekvencijski pretvarač

#### 3.4.1. Princip rada frekvencijskog pretvarača

Frekvencijski pretvarač je elektronički uređaj za promjenu frekvencije i napona napajanja elektromotora. Koristi se za upravljanje brzine i momenta izmjeničnog elektromotora mijenjanjem frekvencije ulazne električne energije. Tipično, frekvencijski pretvarač sastoji se od četiri glavna dijela: ispravljača, istosmjernog međukruga, kočnog sklopa i izmjenjivača.[9][10] Shema frekvencijskog pretvarača s označenim dijelovima prikazana je na Slici 3.7.



Slika 3.7. Shema neizravnog frekvencijskog pretvarača.[9]

Pretvarač je na ulazu spojen na izmjenični izvor (elektroenergetsku mrežu), te se preko diodnog ispravljača izmjenična struja pretvara u istosmjernu. Dobivena istosmjerna struja valovitog je oblika, zbog čega se izgladuje, to jest, smanjuje se njezina valovitost preko istosmjernog međukruga (kondenzatora). Kočni sklop smanjuje naponsko naprezanje kondenzatora tako da se dio struje zatvara preko otpornika. Na kraju, izmjenjivač izgladenu istosmjernu struju pretvara natrag u izmjeničnu s promijenjenom frekvencijom i naponom. Time je, kao što je navedeno u ranijem potpoglavlju, moguće upravljati brzinom vrtnje motora korištenjem skalarne metode.[9][10]

### 3.4.2. Yaskawa J1000

U ovom je radu korišten frekvencijski pretvarač Yaskawa J1000 prikazan na Slici 3.8.



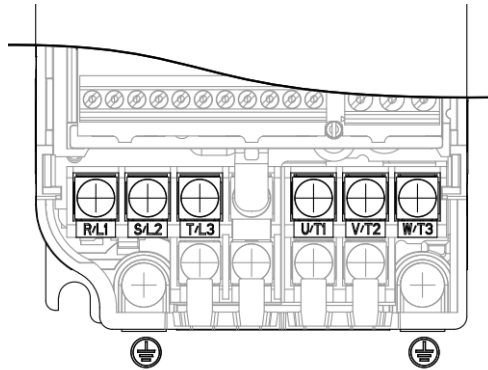
Slika 3.8. Yaskawa J1000 frekvencijski pretvarač.[11]

U nastavku u Tablici 3.2. su navedeni podaci s natpisne pločice frekvencijskog pretvarača.

Tablica 3.2. Podaci frekvencijskog pretvarača.

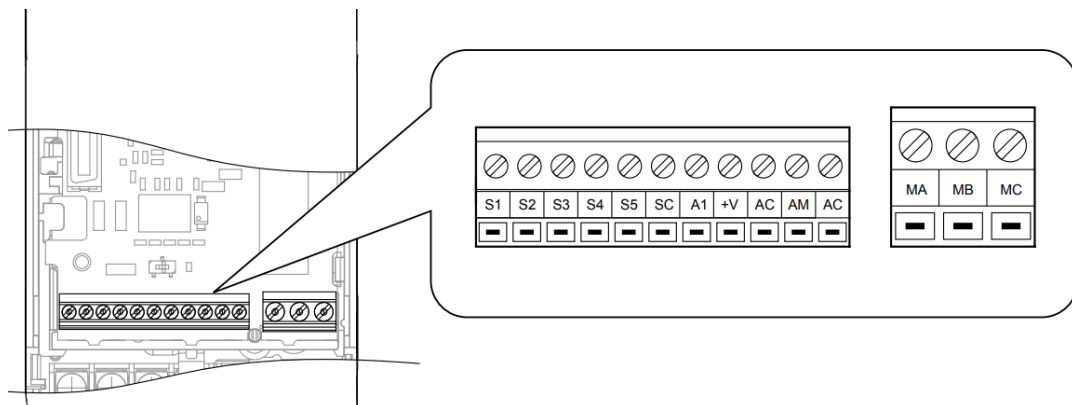
Nazivna snaga motora ( <i>MAX APPLI.MOTOR</i> )	0.37kW/0.25kW
Ulazni broj faza	1
Ulazni napon	200-240V
Ulazna frekvencija	50/60Hz
Ulazna struja	3.6A/2.8A
Izlazni broj faza	3
Izlazni napon	0-240V
Izlazna frekvencija	0-400Hz
Izlazna struja	1.9A/1.6A

Yaskawa J1000 sadrži dva strujna kruga: energetske krug i upravljački krug. Energetski krug, čiji priključci su prikazani na Slici 3.9., služi za napajanje asinkronog motora. Frekvencijski pretvarač automatski pretvara jednofaznu struju u trofaznu kako bi se osigurao ispravan rad motora.



Slika 3.9. Shema priključaka u energetske krugu.[12]

Drugi krug, upravljački krug, služi za upravljanje frekvencijskim pretvaračem, to jest, za upravljanje motorom. Yaskawa J1000 sadrži 14 priključaka za upravljanje (Slika 3.10.), no u svrhu ovog rada se koriste 7 priključaka, čije funkcije su opisane u Tablici 3.3..



Slika 3.10. Shema priključaka u upravljačkom krugu.[12]

Tablica 3.3. Funkcija korištenih priključaka u upravljačkom krugu.

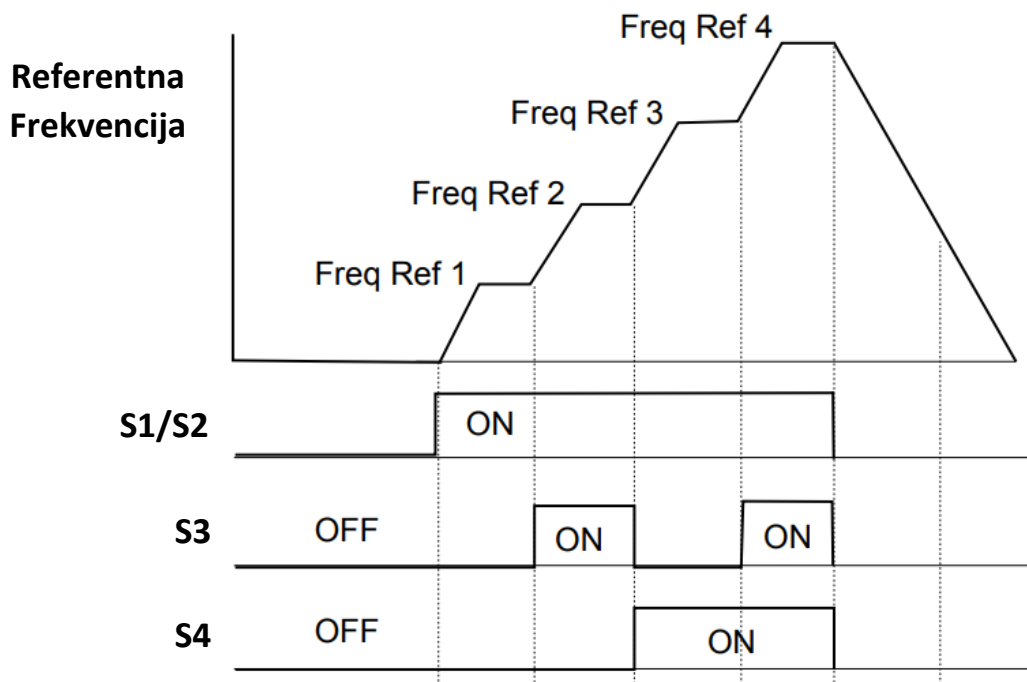
Priključak	Funkcija
S1-S4	Višefunkcionalni ulazi; željenu funkciju pojedinog priključka odabiremo na samom frekvencijskom pretvaraču
S1	Pokretanje/zaustavljanje motora unaprijed
S2	Pokretanje/zaustavljanje motora unazad
S3	Pozivanje referentne frekvencije 2
S4	Pozivanje referentne frekvencije 3
SC	Zajednički priključak (engl. <i>Common Terminal</i> ) za višefunkcionalne ulaze
AM	Analogni izlaz (mjerjenje izlazne frekvencije)
AC	Zajednički priključak za analogni izlaz

Priključci S1 i S2 koriste se za pokretanje i zaustavljanje motora (ako je ulaz 1 motor se pokreće, te ako je ulaz 0 motor se zaustavlja), S3 i S4 za upravljanje brzinom vrtnje motora, te AM za mjerjenje trenutne frekvencije motora.

Upravljanje brzinom motora postiže se tako da se na samom frekvencijskom pretvaraču unaprijed definiraju željene vrijednosti referentnih frekvencija. Zatim, dovođenjem signala na određeni ulaz, pretvarač poziva referentnu frekvenciju koja je pridružena tom ulazu, te ju šalje motoru. U ovom se radu koriste dva ulaza za kontrolu brzine, što daje ukupno četiri različite brzine kojom se motor može rotirati.

Princip rada upravljanja brzinom je sljedeći: ako se signal dovede na S1 ili S2 (pokretanje motora), bez da se dovede na S3 ili S4, frekvencijski pretvarač će motoru poslati *referentnu frekvenciju 1*. Ako se signal također dovede na priključak S3, pretvarač će poslati *referentnu frekvenciju 2*, te isto tako ako se dovede na S4, poslat će *referentnu frekvenciju 3*. U slučaju da se signal dovede na S3 i S4 istovremeno, pretvarač će motoru poslati *referentnu frekvenciju 4*.

Grafički prikaz upravljanja brzinom vrtnje motora pomoću Yaskawe J1000 prikazan je na Slici 3.11.



Slika 3.11. Upravljanje brzinom vrtnje putem upravljačkog kruga.[12]

### 3.4.3. Parametriranje frekvencijskog pretvarača

Ako želimo da frekvencijski pretvarač obavlja dosad navedene funkcije, ponajprije je potrebno podesiti njegove parametre. Parametri se postavljaju preko samog frekvencijskog pretvarača. Budući da Yaskawa J1000 sadrži više od 100 različitih parametara, u nastavku će biti opisani samo promijenjeni parametri, dok svi ostali parametri su ostavljeni na njihovim tvorničkim postavkama.

Tablica 3.4. Postavljeni parametri frekvencijskog pretvarača.[12]

Parametar	Opis	Odabrana funkcija/veličina
b1-02	Odabir izvora unosa naredbe za pokretanje	1 (Digitalni ulazni terminali)
b1-03	Odabir metode zaustavljanja kada se naredba za pokretanje ugasi	1 (Zaustavljanje inercijom)

C1-01	Vrijeme za ubrzanje od 0 do referentne frekvencije	1.0 (sekunda)
d1-1	Referentna frekvencija 1	12.5 (Hz)
d1-2	Referentna frekvencija 2	25.0 (Hz)
d1-3	Referentna frekvencija 3	37.5 (Hz)
d1-4	Referentna frekvencija 4	50.0 (Hz)
H1-01	Funkcija ulaza S1	40 (Pokretanje unaprijed)
H1-02	Funkcija ulaza S2	41 (Pokretanje unazad)
H1-03	Funkcija ulaza S3	3 (Pozivanje referentne frekvencije 2)
H1-04	Funkcija ulaza S4	4 (Pozivanje referentne frekvencije 3)
H1-05	Funkcija ulaza S5	F (Ne koristi se)
H4-01	Odabir podatka za izlaz kroz analogni izlaz AM.	102 (Izlazna frekvencija)

### 3.5. Reduktor

Reduktor je mehanički uređaj koji se koristi za smanjenje brzine vrtnje i povećanje okretnog momenta osovine, a da se pritom brzina vrtnje motora ne mijenja. Reduktor se ugrađuje između motora i radnog dijela stroja, te se sastoji od niza zupčanika ili drugih prijenosnih elemenata koji smanjuju broj okretaja ulazne osovine te prenose veći okretni moment na izlaznu osovinu.[13] Prijenosni omjer reduktora je omjer broja okretaja ulazne osovine prema broju okretaja izlazne osovine unutar reduktora. Drugim riječima, prijenosni omjer pokazuje koliko rotacija ulazna osovina mora izvršiti da bi izlazna osovina izvršila jednu rotaciju. U ovom je radu korišten reduktor s prijenosnim omjerom 42, koji možemo vidjeti na Slici 3.12..

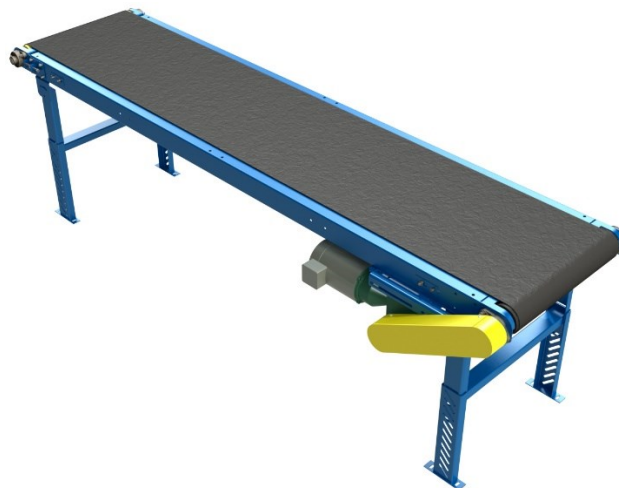




*Slika 3.12. Reduktor spojen na asinkroni motor.*

### **3.6. Pokretna traka**

Posljednja komponenta u sustavu je pokretna traka. Pokretna traka je mehanički uređaj koji se koristi za prijenos predmeta i materijala. Sastoji se od trake koja se kreće po valjcima koji su pogonjeni motorom.

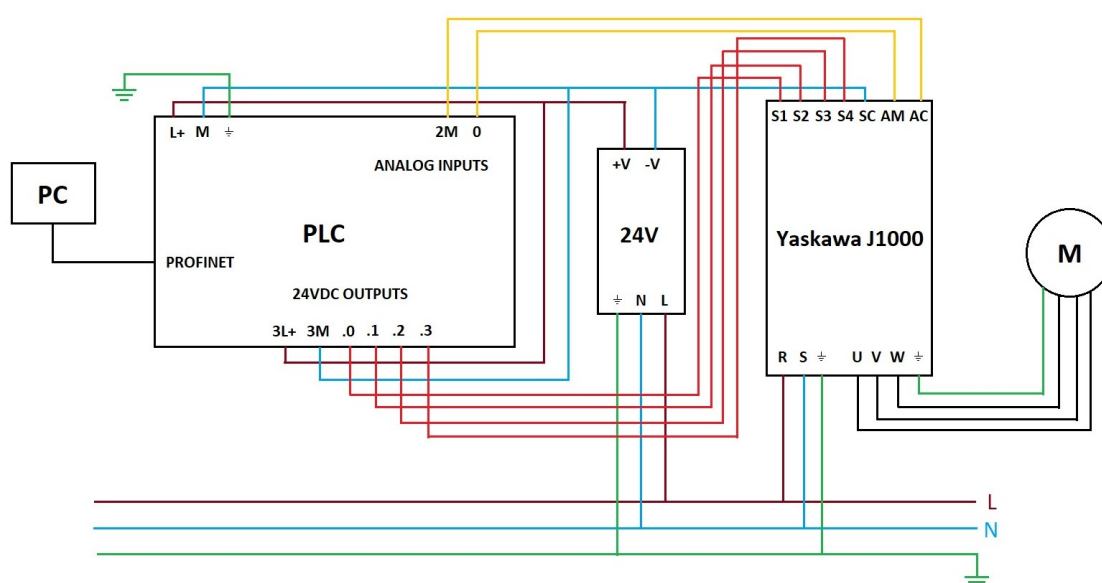


*Slika 3.13. Pokretna traka.[14]*

## 4. OPIS SUSTAVA

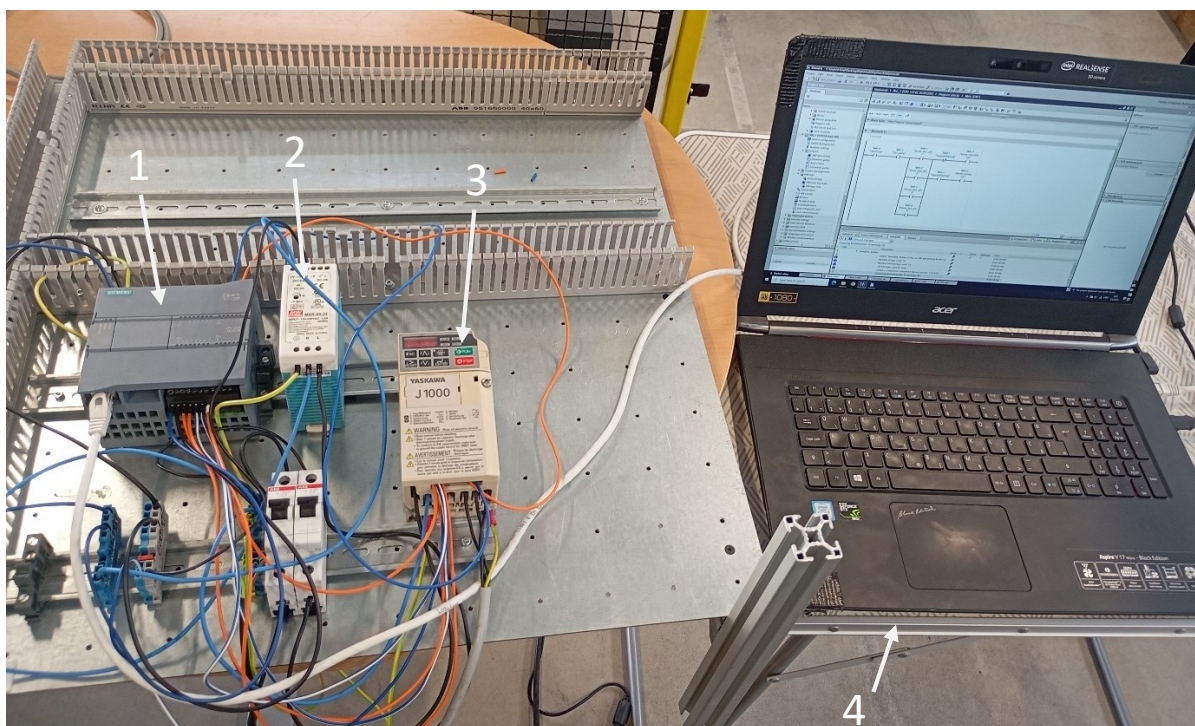
### 4.1. Povezivanje komponenti

Dosad navedene komponente potrebno je međusobno povezati da bismo dobili konačan sustav upravljanja pokretnom trakom. U nastavku na Slici 4.1. prikazana je cjelovita shema povezivanja, odnosno, ožičenja komponenti, kao i detaljan opis pojedinog dijela.



Slika 4.1. Dijagram ožičenja upravljačkog sustava.

Ponajprije, osiguramo napajanje asinkronog motora preko energetskog kruga frekvencijskog pretvarača. Fazu  $L$ , nulu  $N$ , i uzemljenje spojimo na ulazne priključke  $R$ ,  $S$ , i uzemljenje respektivno, te izlazne priključke  $U$ ,  $V$ ,  $W$  i uzemljenje spojimo na motor. Izvor zatim spojimo na 24-voltni modul koji će služiti za napajanje PLC-a. Pozitivni izlaz modula  $+V$  spojimo na PLC priključke  $L+$  i  $3L+$ , te negativni izlaz  $-V$  spojimo na priključke  $M$  i  $3M$ , čime se osigura rad PLC ulaza i izlaza. Digitalne PLC izlaze  $.0$ - $.3$ , pomoću kojih ćemo upravljati motorom, spojimo na višefunkcionalne ulaze frekvencijskog pretvarača,  $S1$ - $S4$ , te priključak  $SC$  spojimo na natrag na  $3M$ . Zatim, analogni izlaz iz frekvencijskog pretvarača,  $AM$ , spojimo na analogni ulaz PLC-a,  $0$ , te  $AC$  spojimo na  $2M$ . Na kraju, s Ethernet kabelom povežemo računalo i PLC, tako da se uspostavi komunikacija između dva uređaja putem Profinet protokola.



Slika 4.2. Sustav upravljanja s označenim komponentama: 1) PLC, 2) Napajanje 24 V, 3) Yaskawa J1000 frekvenijski pretvarač, 4) Računalo.



Slika 4.3. Sustav upravljanja s označenim komponentama: 1) Pokretna traka, 2) Reduktor, 3) Asinkroni motor.

## 4.2. Opis rada sustava

Ideja ovog rada je da preko HMI-a, odnosno, računala korisnik odabere željen način rada sustava (otvorena ili zatvorena petlja), brzinu pokretne trake, te željeni pomak ili poziciju trake. Računalo odabrane vrijednosti šalje PLC-u koji aktivira određene izlaze koji su spojeni na ulaze frekvencijskog pretvarača. PLC izlaz .0 spojen je na ulaz *S1*, te se preko njega asinkroni motor pokreće i zaustavlja. Motor je preko reduktora spojen na pokretnu traku, tako da pokretanjem i zaustavljanjem motora, pokrećemo i zaustavljamo pokretnu traku. Ako je izlaz aktiviran, motor, to jest, traka se giba, te ako nije, traka miruje. Isto vrijedi i za PLC izlaz .1, samo što se traka giba u suprotnom smjeru. PLC izlazi .2 i .3 spojeni su na ulaze *S3* i *S4*, tako da oni služe za upravljanje brzinom pokretne trake. Detaljniji opis upravljanja brzinom dan je u potpoglavlju 3.4..

U slučaju da korisnik odabere upravljanje u otvorenoj petlji, sustav će raditi bez povratne veze. Korisniku je omogućeno da preko HMI-a odabere željenu brzinu koji aktivira odgovarajuće PLC izlaze .2 i .3. Korisnik zatim odabire željeni pomak, to jest, udaljenost koju želi da pokretna traka prijeđe, te smjer. Pritiskom na za to određeno HMI tipkalo, korisnik aktivira izlaz .0 ili .1, čime pokreće pokretnu traku. Na temelju odabrane brzine i pomaka, PLC će izračunati potrebno vrijeme gibanja trake prije zaustavljanja. Korisniku je također omogućeno da u bilo kojem trenutku sam zaustavi traku.

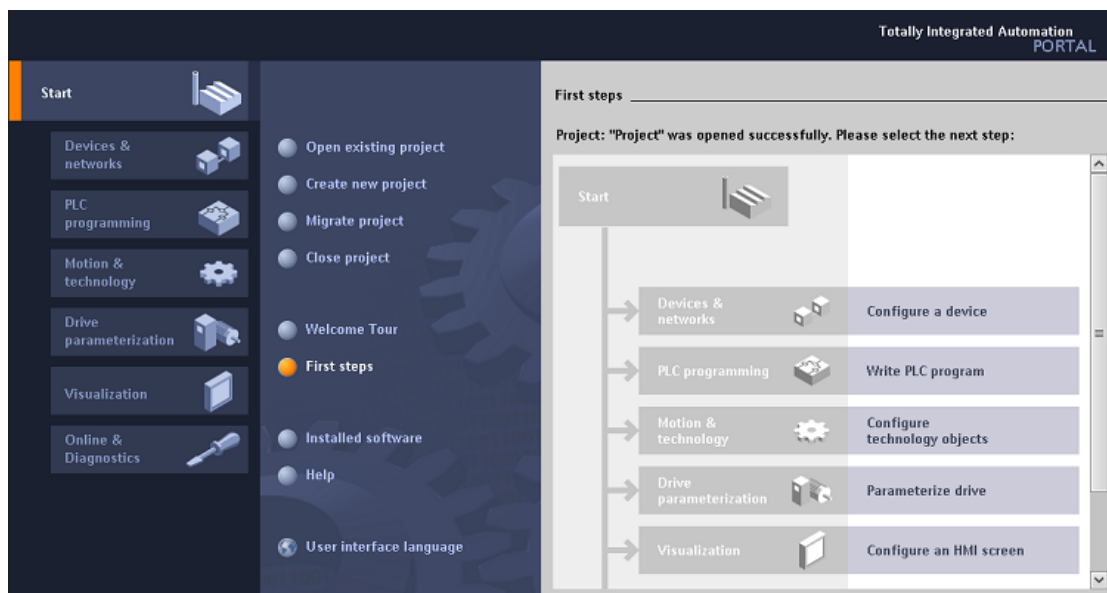
U zatvorenoj petlji, sustav će preko povratne veze dobivati povratne informacije o stvarnom stanju izlaza sustava. Kao povratna veza koristi se izlaz frekvencijskog pretvarača *AM* koji je spojen na PLC ulaz 0. U zatvorenoj petlji, korisniku je umjesto pomaka omogućeno da odabere željenu poziciju. Sustav u zatvorenoj petlji radi tako da izlaz *AM* PLC-u šalje trenutnu vrijednost frekvencije koju pretvarač šalje motoru. Na temelju primljene frekvencije, PLC izračunava trenutnu brzinu pokretne trake, a time i prijeđeni put, tako da sustav u svakom trenutku zna trenutnu poziciju pokretne trake. Traka se giba sve dok stvarna pozicija ne postane jednaka željenoj poziciji, u kojem slučaju se traka zaustavlja. Budući da sustav ne posjeduje senzore za detekciju objekta na pokretnoj traci (npr. senzor blizine), sustav uvijek smatra da je početna pozicija objekta na početku trake.

Sam programski kod za upravljanje sustavom i HMI sučelje opisani su u poglavlju 6.

## 5. TIA PORTAL

Kao što je spomenuto u ranijem poglavlju, u svrhu razvoja programske aplikacije za upravljanje pokretnom trakom korišten je program TIA Portal. TIA Portal je softverska platforma koja omogućuje programiranje, nadziranje i održavanje automatiziranih sustava i procesa. Neke od značajka i alata koje TIA Portal uključuje su:

- Programiranje PLC-a
- Dizajniranje i programiranje HMI panela
- Simulacija i testiranje programa
- Dijagnostika za pronalaženje pogrešaka u sustavu.[15]

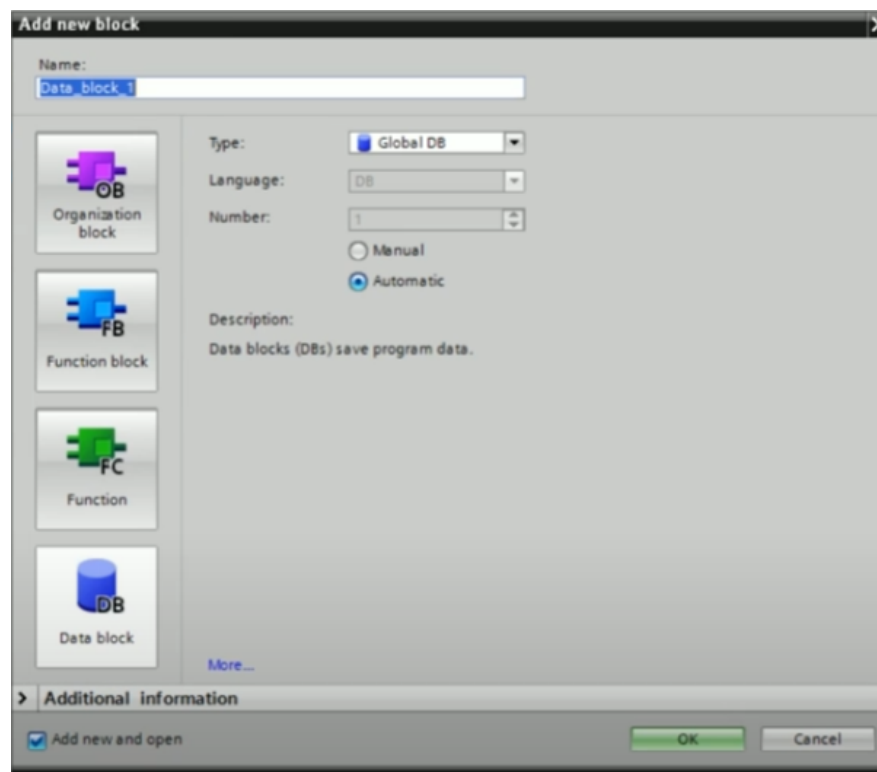


Slika 5.1. TIA Portal početni zaslon.

Programiranje u TIA Portalu uključuje korištenje različitih blokova za stvaranje logike upravljanja i upravljanje podacima. Četiri glavna tipa programskih blokova u TIA Portalu su:

- Organizacijski blokovi (engl. *Organization Blocks*) ili OB
- Funkcijski blokovi (engl. *Function Blocks*) ili FB
- Funkcije (engl. *Functions*) ili FC
- Podatkovni blokovi (engl. *Data Blocks*) ili DB

Organizacijski blokovi, OB, određuju strukturu korisničkog programa, gdje se cijeli korisnički program ciklički poziva iz OB1. Funkcijski blokovi, FB, su dijelovi koda koji se mogu pozivati, te izvršavaju određene zadatke i vraćaju rezultat. Slični su funkcijama u drugim programskim jezicima, uz to da posjeduju vlastitu memoriju, odnosno, mogućnost zadržavanja informacija o stanju. Funkcije, FC, imaju istu ulogu kao FB, ali ne posjeduju vlastitu memoriju, zbog čega se koriste za zadatke koji ne zahtijevaju trajnu pohranu podataka. Na kraju, podatkovni blokovi, DB, se koriste za pohranjivanje podataka koje program koristi.[16]



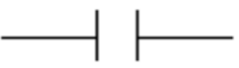
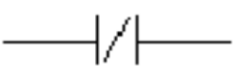
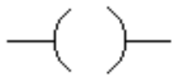



Slika 5.2. Odabir programskog bloka u TIA Portalu.

## 5.1. Ljestvičasti dijagram

Ljestvičasti dijagram (engl. *Ladder Diagram*) ili LD programski je jezik koji omogućuje izradu programskog koda pomoću grafičkog dijagrama temeljenog na dijagramima strujnih upravljačkih shema, te se često koristi za programiranje PLC-ova. Pisanje koda u LD temelji se na mreži koja se sastoji od simbola koji predstavljaju instrukcije i upravljačke komponente.[17]

U Tablici 5.1. navedene su neke od osnovnih operacija u LD.

Tablica 5.1. Osnovni simboli i operacije u ljestvičastom dijagramu.[17]

Simbol	Naziv	Funkcija
	Radni kontakt (engl. <i>Normally Open Contact</i> )	Provodi ako je zatvoren
	Mirni kontakt (engl. <i>Normally Closed Contact</i> )	Provodi ako je otvoren
	Izlazni svitak (engl. <i>Output Coil</i> )	Uključen ako je signal prisutan; isključen ako nema signala
	Invertirani izlazni svitak (engl. <i>Inverted Output Coil</i> )	Uključen ako nema signala; isključen ako je signal prisutan
	SET	Uključen ako je signal prisutan, te ostaje uključen dok nema signala
	RESET	Isključen ako je signal prisutan, te ostaje isključen dok nema signala

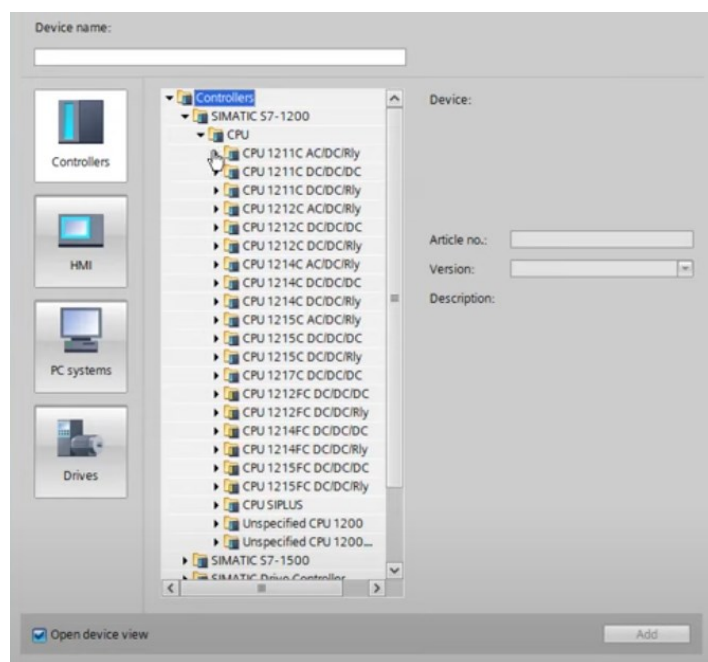
Osim navedenih operacija, u ovom radu korišteni su i sljedeći funkcijski blokovi:

- *TON* - Odgađa aktivaciju izlaza za unaprijed određeno vrijeme nakon što je ulazni signal aktiviran.
- *P\_TRIG* - Ako se njegovo stanje promjeni iz logičke nule u jedinicu, provede signal kroz jednu iteraciju programa.
- *CONV* – Pretvara jednu vrstu podataka u drugi.
- *TP* - Generira impulsni signal određenog trajanja kada je ulazni signal aktiviran.
- *MOVE* – Kopira vrijednost iz jedne varijable u drugu.
- *MUL* – Množi dvije varijable.
- *DIV* - Dijeli dvije varijable.
- *CALCULATE* – Izračunava vrijednost željenog matematičkog izraza unesenog u blok.

## 6. PROGRAMSKA REALIZACIJA

### 6.1. Uspostava komunikacije uređaja

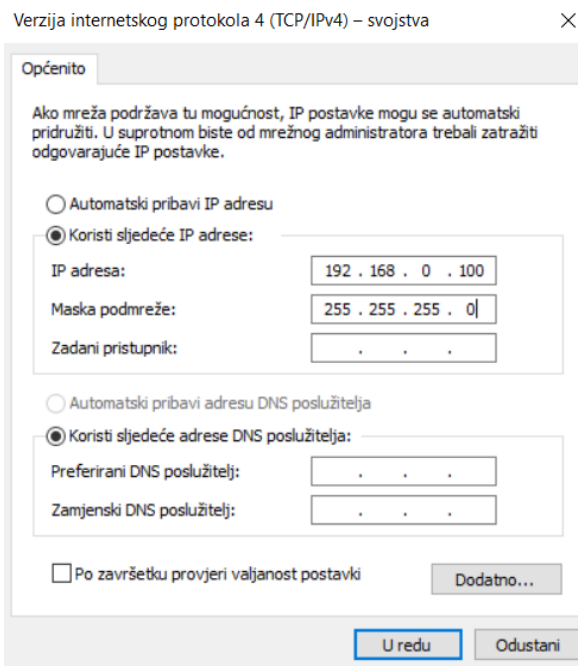
Prije nego što krenemo s pisanjem samog programskog koda, potrebno je u TIA Portalu konfigurirati uređaje koje ćemo koristiti, te uspostaviti komunikaciju između njih. Ponajprije u projekt, pomoću opcije „Add new device“, dodamo željeni PLC i HMI modul. Točna serija PLC modula u ovom radu je „SIMATIC S7-1200, CPU 1214C DC/DC/DC, 6ES7214-1AG40-0XB0, Version 4.2“. Budući da se u ovom radu ne koristi fizički HMI, nego simulacija unutar samog programa, možemo odabrati bilo koji HMI panel iz izbornika.



Slika 6.1. Izbornik za dodavanje uređaja.

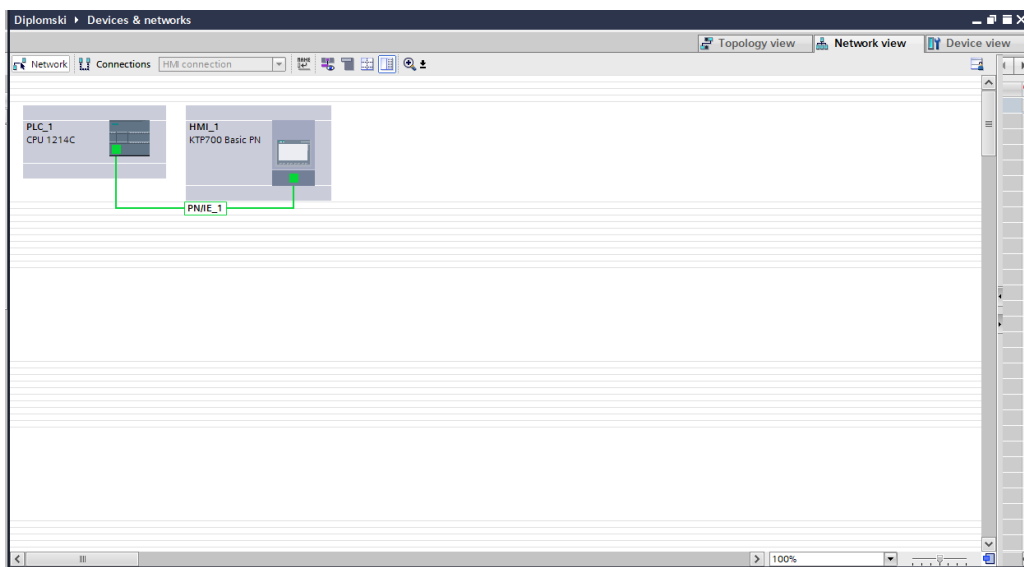
Nakon što dodamo uređaje, potrebno je konfigurirati IP adrese. PLC-u i HMI-u automatski se dodjele IP adrese *192.168.0.1* i *192.168.0.2* respektivno, te da bismo mogli uspostaviti komunikaciju između računala i PLC-a, potrebno je da imaju istu podmrežu (engl. *Subnet*). To postignemo tako da na računalu promijenimo IP adresu IPv4 protokola. Prva tri polja IP adrese trebaju biti jednaka adresi PLC-a da bi se uspostavila ista podmreža, te je potrebno da zadnje polje bude drugačije, kao što je prikazano na Slici 6.2.





Slika 6.2. Promjena IP adrese IPv4.

Na kraju, u TIA Portalu u prozoru „*Network view*“ povežemo PLC i HMI, čime se uspostavi njihova komunikacija.



Slika 6.3. Povezivanje PLC-a i HMI-a.

## 6.2. Programski kod

Nakon što konfiguriramo i komunikacijski povežemo korištene uređaje, slijedi pisanje samog programskog koda koji će upravljati pozicioniranjem trake. Prvi korak je definiranje PLC tagova koje ćemo koristiti u programskom kodu. U TIA Portalu, tagovi se koriste za označavanje varijabli koje se koriste u samom projektu. Tagovi, odnosno, varijable mogu biti različitih tipova podataka, te su u ovom radu korišteni podaci prikazani i Tablici 6.1.

Tablica 6.1. Opis korištenih tipova podataka.[18]

Naziv	Veličina	Raspon	Opis
BOOL	1 bit	[0, 1] (True/False)	1-bitna binarna vrijednost
WORD	16 bitova	[16#0000, 16#FFFF]	16-bitna binarna vrijednost
INT	16 bitova	[-32768, 32767]	16-bitni broj s predznakom s nepomičnom decimalnom točkom
DINT	32 bitova	[-2147483648, 2147483647]	32-bitni broj s predznakom s nepomičnom decimalnom točkom
REAL	32 bitova	$[-3.402823 \cdot 10^{38}, 3.402823 \cdot 10^{-38}]$	32-bitni broj s predznakom s pomičnom decimalnom točkom

Osim tipa podataka, kod definiranja tagova potrebno je također odrediti i njihovu adresu, te dali je podatak ulazni, izlazni ili memorijski. Ulazni podaci (označeni prefiksom *I*) predstavljaju signale koje PLC prima na fizikalni ulaz od vanjskih uređaja, dok izlazni podaci (označeni prefiksom *Q*) predstavljaju signale koje PLC šalje vanjskim uređajima. Adrese ulaza i izlaza odgovaraju adresama koje su označene na PLC modulu. Na kraju, memorijski podaci (označeni prefiksom *M*) su unutarne varijable u PLC-u koje se koriste za privremeno pohranjivanje podataka unutar programa, te nisu izravno povezane s vanjskim ulazima ili izlazima.

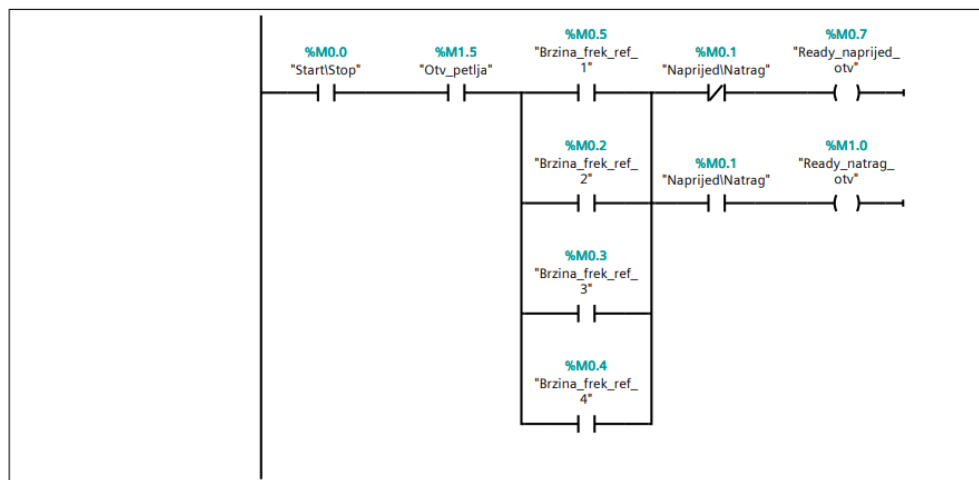
Diplomski ▶ PLC_1 [CPU 1214C DC/DC/DC] ▶ PLC tags ▶ PLC_tags [55]								
PLC_tags								
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	StartStop	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	StartStop_naprijed_out	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	NaprijedNatrag	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	StartStop_natrag_out	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Brzina_frek_ref_2	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Brzina_frek_ref_2_out	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Brzina_frek_ref_3	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Brzina_frek_ref_3_out	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Brzina_frek_ref_4	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Brzina_frek_ref_1	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	StartStop_HMI	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Pozicija	Int	%MW4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Ready_naprijed_otv	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Ready_natrag_otv	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Frekvencija_in	Word	%IW64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Trig_startstop	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	a	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	b	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	trig_zat	Bool	%M1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	s_zat	Real	%MD8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	s_zat_int	Int	%MW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Otv_petlja	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Zat_petlja	Bool	%M1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	Brojilo	Bool	%M1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	Ready_zat	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	Error	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	v_lin_zat	Int	%MW22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Slika 6.4. Definirani PLC tagovi.

Programski kod možemo podijeliti u dva dijela: dio koji se odnosi na otvorenu petlju i dio koji se odnosi na zatvorenu petlju. Prvo će biti objašnjen dio koji se odnosi na otvorenu petlju.

### 6.2.1. Upravljanje u otvorenoj petlji

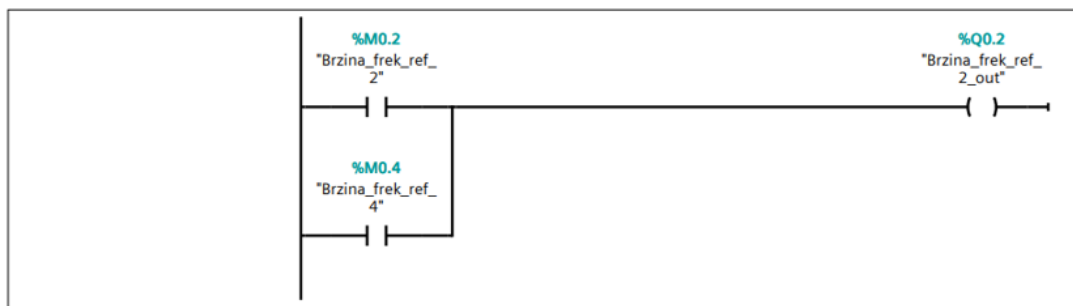
Network 1:



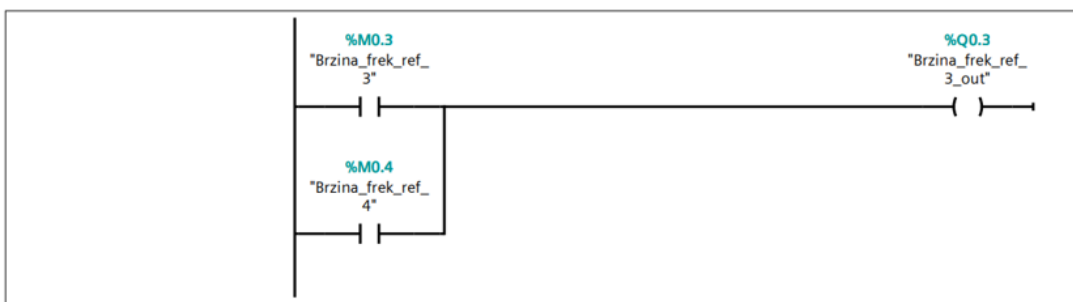
Slika 6.5. Network 1.

U *Networku 1* navedeni su uvjeti koji trebaju biti ispunjeni za pokretanje pokretne trake u otvorenoj petlji. Sami uvjeti se ispunjavaju preko HMI sučelja pomoću kojeg upravljamo memorijskim varijablama, te će način na koji se traka pokreće biti objašnjen u sljedećem potpoglavlju gdje se opisuje rad HMI-a.

#### Network 4:



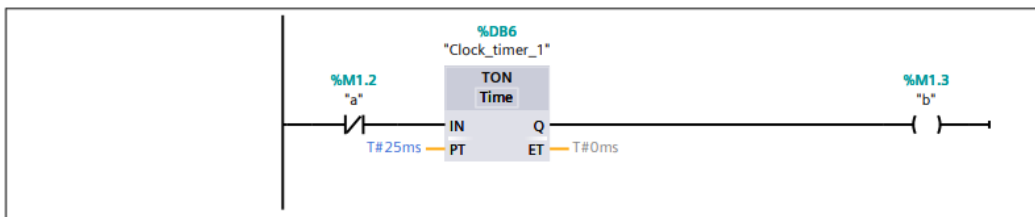
#### Network 5:



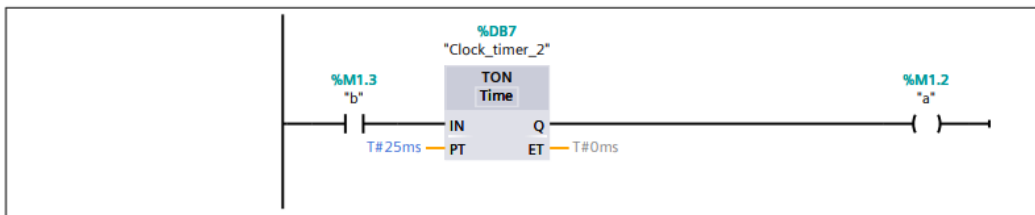
*Slika 6.6. Network 4 i Network 5.*

*Network 4* i *Network 5* služe za kontrolu brzine. Kao što je opisano u poglavlju s frekvencijskim pretvaračem, brzina motora se mijenja dovođenjem signala na priključke *S3* i *S4*, koji su u našem slučaju povezani s PLC izlazima *%Q0.2* i *%Q0.3*. Ponovno, biranje brzine se čini preko HMI sučelja koji upravlja varijablama *%M0.2*, *%M0.3* i *%M0.3*.

**Network 6:**



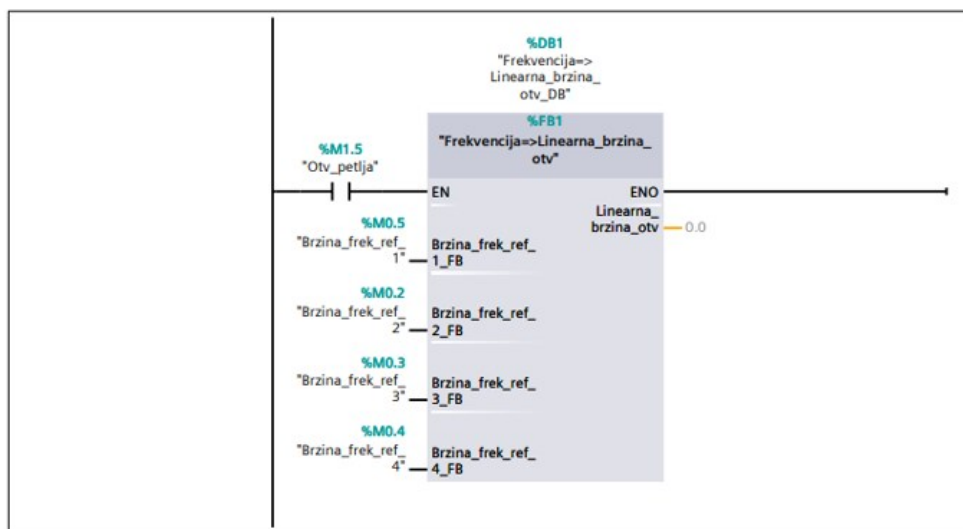
**Network 7:**



*Slika 6.7. Network 6 i Network 7.*

*Network 6* i *Network 7* služe za periodično uključivanje i isključivanje varijable 'b', koja će nam kasnije poslužiti za izračunavanje prijednog puta, odnosno, trenutne pozicije trake. Postavljeno vrijeme odgađanja na *TON* tajmerima iznosi 25 milisekundi, što znači da se varijabla 'b' uključuje svakih 50 milisekundi, to jest, treperi frekvencijom od 20 Hz.

**Network 8:**



*Slika 6.8. Network 8.*

U *Networku 8* pozivamo funkcijski blok koji služi za preračunavanje frekvencije motora u linearnu brzinu pokretne trake. U nastavku je opisan princip rada bloka.

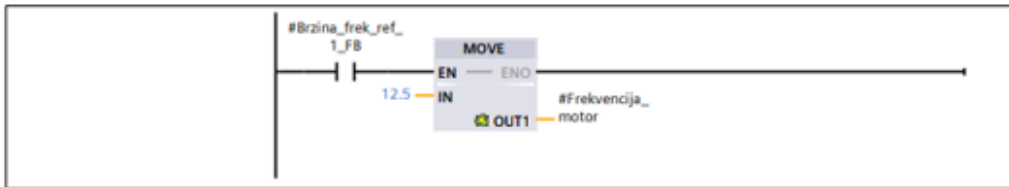
Frekvencija=>Linearna_brzina_otv									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Set-point	Supervision	Comment
▼ Input									
Brzina_frek_ref_1_FB	Bool	false	Non-retain	True	True	True	False		
Brzina_frek_ref_2_FB	Bool	false	Non-retain	True	True	True	False		
Brzina_frek_ref_3_FB	Bool	false	Non-retain	True	True	True	False		
Brzina_frek_ref_4_FB	Bool	false	Non-retain	True	True	True	False		
▼ Output									
Linearna_brzina_otv	Real	0.0	Non-retain	True	True	True	False		
InOut									
Static									
▼ Temp									
Frekvencija_motor	Real								
Constant									

Slika 6.9. Varijable u „Frekvencija=>Linearna\_brzina\_otv“ funkcijskom bloku

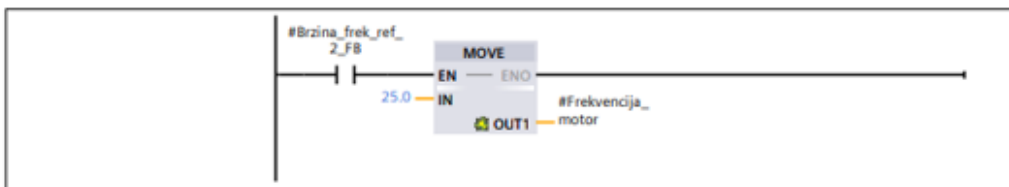
Prvo definiramo varijable koje će se koristiti funkcijskom bloku. Funkcijski blok može sadržavati 6 različitih tipova varijabli:

- Input (Ulaz) - Varijable koje se izvane unose u funkcijski blok.
- Output (Izlaz) - Varijable koje funkcijski blok daje na svom izlazu.
- InOut (Ulaz-Izlaz) - Varijable koje se mogu koristiti i kao ulazi i kao izlazi.
- Static - Varijable koje zadržavaju svoju vrijednost između različitih poziva funkcijskog bloka.
- Temp - Privremene varijable koje se koriste unutar funkcijskog bloka i koje postoje samo za vrijeme izvršavanja bloka.
- Const - Varijable koje služe kao konstantne ulazne vrijednosti koje se ne mogu mijenjati tijekom izvršavanja bloka.

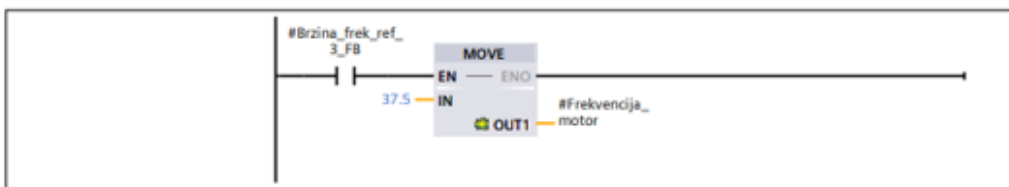
**Network 1:**



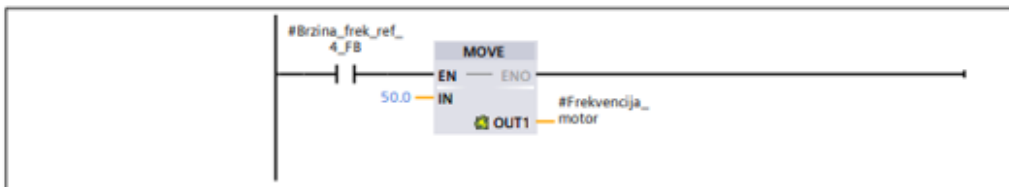
**Network 2:**



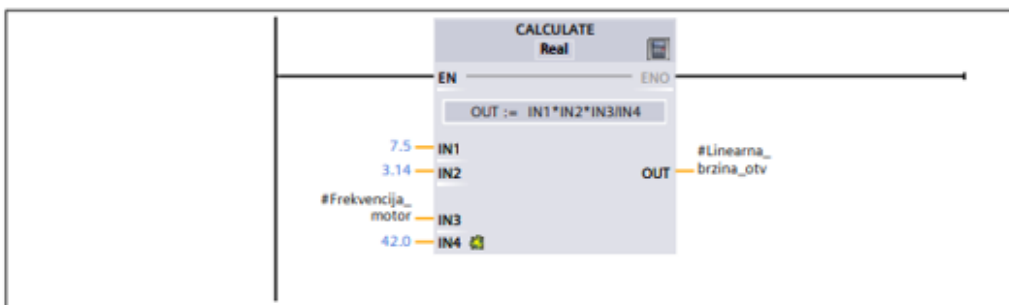
**Network 3:**



**Network 4:**



**Network 5:**



Slika 6.10. Funkcijski blok „Frekvencija=>Linerana\_brzina\_otv“.

FB na ulazu prima jednu od četiri unaprijed odabranih frekvencija, koju zatim preko bloka *MOVE* dovodi do bloka *CALCULATE* koji preračunava frekvenciju u linearnu brzinu. Da bismo dobili jednadžbu za preračunavanje, frekvenciju je ponajprije potrebno pretvoriti u kutnu brzinu prema izrazu:

$$\omega = 2 \cdot \pi \cdot f, \quad (6.1)$$

gdje je:  $\omega$  kutna brzina u rad/s,  $f$  frekvencija u Hz. Navedeni je izraz zatim potrebno uvrstiti u jednadžbu za linearnu brzinu koja glasi:

$$v = \omega \cdot r, \quad (6.2)$$

gdje je:  $v$  linearna brzina pokretne trake u cm/s,  $r$  radijus valjka pokretne trake u cm (3,75 cm). Uvrštavanjem dobijemo izraz:

$$v = 2 \cdot \pi \cdot f \cdot 3,75 = 7,5 \cdot \pi \cdot f \quad (6.3)$$

Između motora i pokretne trake nalazi se reduktor s omjerom od 42. To znači da će se valjak trake okretati 42 puta sporije od samog motora, zbog čega je jednadžbu (6.3) potrebno podijeliti s omjerom, čime dobijemo konačan izraz:

$$v = \frac{7,5 \cdot \pi \cdot f}{42} \quad (6.4)$$

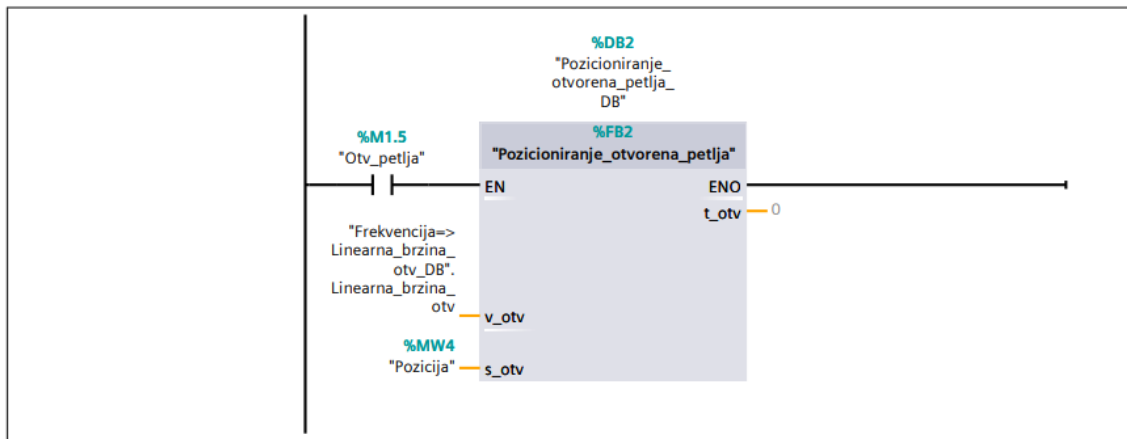
Uvrštavanjem unaprijed određenih frekvencija  $f$  iz Tablice 3.2. dobijemo slijedeće vrijednosti za linearne brzine pokretne trake:

*Tablica 6.2. Vrijednosti linearnih brzina za odgovarajuću frekvenciju.*

Frekvencija [Hz]	Linearna brzina [cm/s]
12.5	7,01
25	14,02
37.5	21,03
50	28,04



## Network 9:



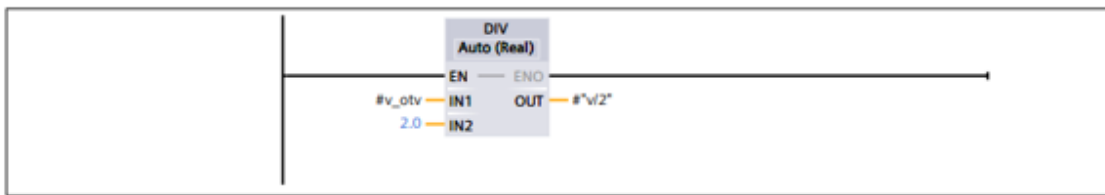
Slika 6.11. Network 9.

Natrag u glavnom organizacijskom bloku, u *Networku 9*, pozivamo funkcijski blok koji izračunava potrebno vrijeme gibanja pokretne trake.

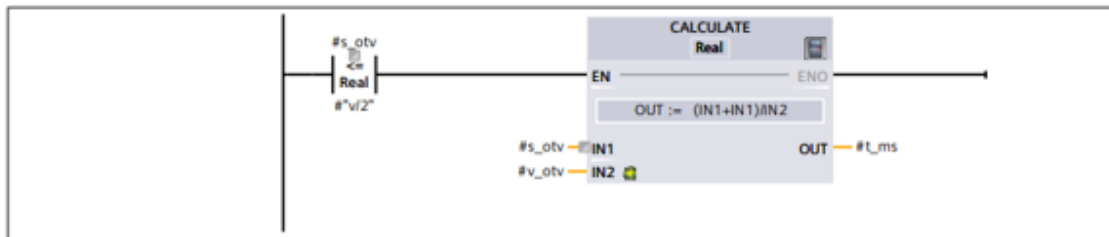
Pozicioniranje_otvorena_petlja									
Name	Data type	Default value	Retain	Accessible from HMI/OP C UA/We b API	Wri ta- ble engi- neer- ing	Visible in HMI	Set- point	Super- vision	Comment
▼ Input									
v_otv	Real	0.0	Non-retain	True	True	True	False		
s_otv	Int	0	Non-retain	True	True	True	False		
▼ Output									
t_otv	DInt	0	Non-retain	True	True	True	False		
InOut									
Static									
▼ Temp									
t_ms	Real								
v/2	Real								
Constant									

Slika 6.12. Variable u „Pozicioniranje\_otvorena\_petlja“ funkcijskom bloku.

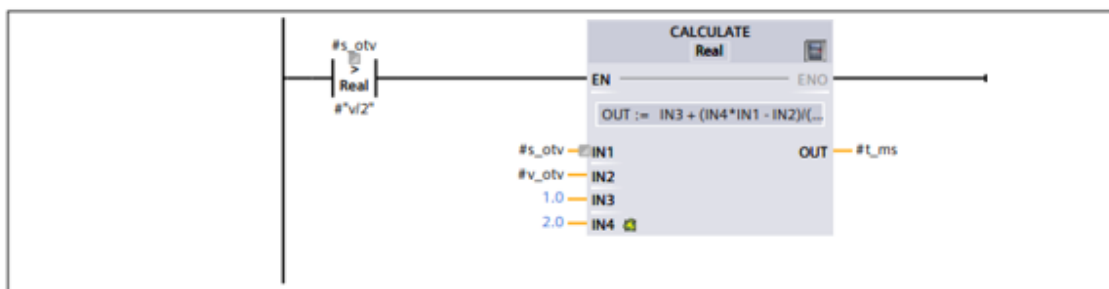
### Network 1:



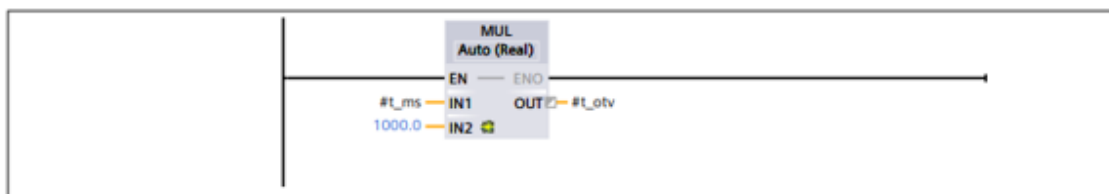
### Network 2:



### Network 3:



### Network 4:



Slika 6.13. Funkcijski blok „Pozicioniranje\_otvorena\_petlja“.

Funkcijski blok kao ulaz prima izračunatu linearnu brzinu pokretne trake, te željenu poziciju. Da bismo precizno izračunali potrebno vrijeme gibanja trake, potrebno je uzet u obzir vrijeme ubrzavanja i vrijeme usporavanja trake. Yaskawa J1000 nudi mogućnost odabira željenog vremena ubrzavanja i usporavanja, odnosno, vrijeme potrebno da frekvencija motora iz nule postigne željenu frekvenciju i obrnuto.

U ovom je radu određeno da vrijeme ubrzanja  $t_{acc}$  uvijek iznosi 1 s (neovisno o konačnoj brzini), dok je pri usporavanju postavljeno da se motor zaustavi pomoću inercije, tako da je vrijeme usporavanja instantno. Budući da su same linearne brzine niske, pri usporavanju neće doći do odmaka željene pozicije pokretne trake. Drugim riječima, pri izračunavanju vremena gibanja trake, potrebno je uzet u obzir samo vrijeme ubrzanja, te vrijeme konstantnog gibanja.

Da bismo odredili potrebno vrijeme gibanja, ponajprije koristimo formulu za udaljenost kod jednolikog ubrzanja:

$$s_{acc} = \frac{1}{2} \cdot a \cdot t_{acc}^2, \quad (6.5)$$

gdje je:  $s_{acc}$  prijeđena udaljenost pri ubrzanju,  $a$  ubrzanje,  $t_{acc}$  vrijeme ubrzanja. Ubrzanje  $a$  zamijenimo općim izrazom za ubrzanje  $a = v_{ref}/t_{acc}$ , te jednadžbu (6.5) možemo zapisati kao:

$$s_{acc} = \frac{v_{ref}}{2} \cdot t_{acc}, \quad (6.6)$$

gdje je:  $v_{ref}$  konačna (referentna) linearna brzina. Zatim, koristimo formulu za prijeđenu udaljenost pri konstantnom gibanju:

$$s_{konst} = v_{ref} \cdot t_{konst}, \quad (6.7)$$

gdje je:  $s_{konst}$  prijeđena udaljenost pri konstantnoj brzini,  $t_{konst}$  vrijeme gibanja konstantnom brzinom. Time dobijemo ukupnu prijeđenu udaljenost  $s$ , koja je jednaka sumi  $s_{acc}$  i  $s_{konst}$ :

$$s = s_{acc} + s_{konst} \quad (6.8)$$

Ako je  $s$  manji od  $v_{ref} \cdot t_{acc}/2$  to znači da se gibanje događa samo za vrijeme ubrzanja. Drugim riječima, ukupna prijeđena udaljenost jednaka je udaljenosti kod ubrzanja, čime dobijemo izraz:

$$s = s_{acc} = \frac{v_{ref}}{2} \cdot t \rightarrow t = \frac{2s}{v_{ref}}, \quad (6.9)$$

gdje je:  $t$  potrebno vrijeme gibanja motora da dođe do željene udaljenosti  $s$ .

Ako je  $s$  veći od  $v_{ref} \cdot t_{acc}/2$  to znači da se gibanje događa i za vrijeme konstantne brzine. Vrijeme gibanja tijekom konstantne brzine iznosi:

$$s = \frac{v_{ref}}{2} \cdot t_{acc} + v_{ref} \cdot t_{konst} \rightarrow t_{konst} = \frac{2s - v_{ref} \cdot t_{acc}}{2v_{ref}}, \quad (6.10)$$

te ako dodamo i vrijeme ubrzanja, dobijemo:

$$t = t_{acc} + t_{konst} = 1 + \frac{2s - v_{ref} \cdot t_{acc}}{2v_{ref}} \quad (6.11)$$

Konačan izraz za potrebno vrijeme gibanja trake glasi:

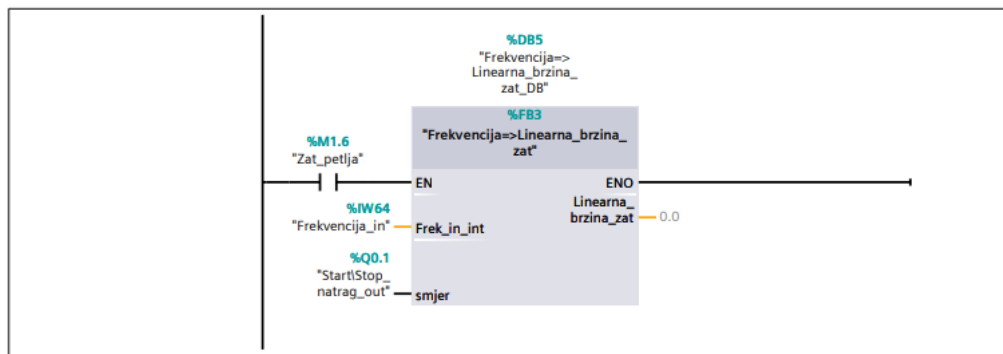
$$t = \begin{cases} \frac{2s}{v_{ref}}, & \text{ako } s \leq \frac{v_{ref}}{2} \cdot t_{acc} \\ 1 + \frac{2s - v_{ref} \cdot t_{acc}}{2v_{ref}}, & \text{ako } s > \frac{v_{ref}}{2} \cdot t_{acc} \end{cases} \quad (6.12)$$

Dobiveno vrijeme dovodimo na  $TP$  blokove (Slika 6.14.) koji aktiviraju PLC izlaz  $\%Q0.0$  ili  $\%Q0.1$ , ovisno o tome želimo li da se traka giba unaprijed ili unatrag. PLC izlazi zatim pokreću pokretnu traku u određenom smjeru, te nakon što vrijeme istekne,  $TP$  blokovi se gase, a i s njima PLC izlazi, te dolazi do zaustavljanja trake.  $TP$  blokovi automatski mjere vrijeme u milisekundama, zbog čega je rezultirajuće vrijeme još potrebno pomnožiti s 1000 prije nego što se dovede na sam blok.



U *Networku 2* su navedeni uvjeti koji trebaju biti ispunjeni za pokretanje pokretne trake u zatvorenoj petlji, slično kao u *Networku 1* za otvorenu petlju. Kao i u otvorenoj petlji, sami uvjeti se ispunjavaju preko HMI-a.

**Network 10:**



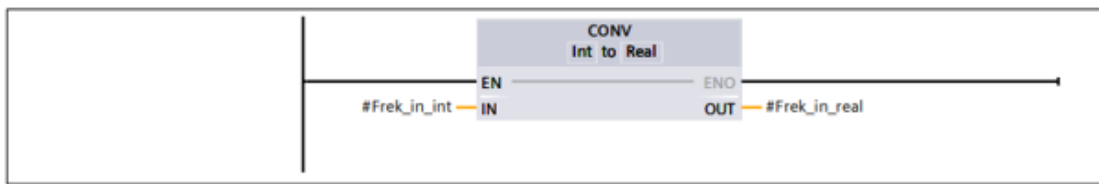
*Slika 6.16. Network 10.*

U *Networku 10* pozivamo funkcijski blok koji preračunava očitane vrijednosti na analognom PLC ulazu u linearnu brzinu pokretne trake.

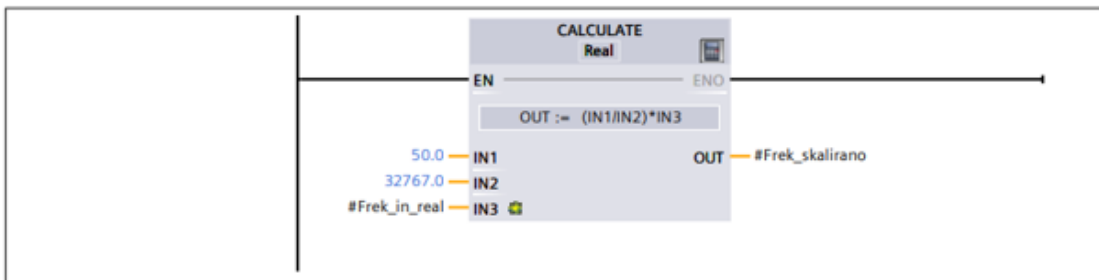
Frekvencija=>Linearna_brzina_zat									
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Set-point	Supervision	Comment
▼ Input									
Frek_in_int	Int	0	Non-retain	True	True	True	False		
smjer	Bool	false	Non-retain	True	True	True	False		
▼ Output									
Linearna_brzina_zat	Real	0.0	Non-retain	False	False	False	False		
InOut									
Static									
▼ Temp									
Frek_in_real	Real								
Frek_skalirano	Real								
Constant									

*Slika 6.17. Varijable u „Frekvencija=>Linearna\_brzina\_zat“ funkcijskom bloku.*

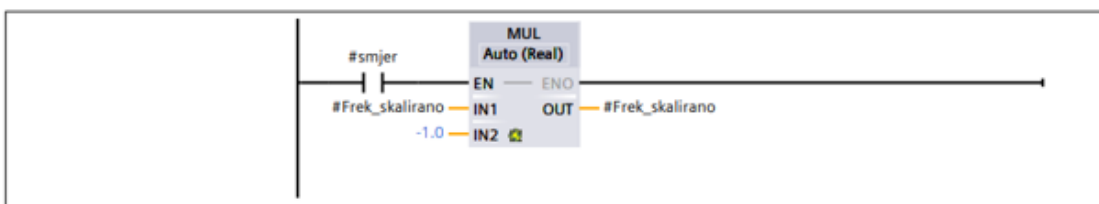
### Network 1:



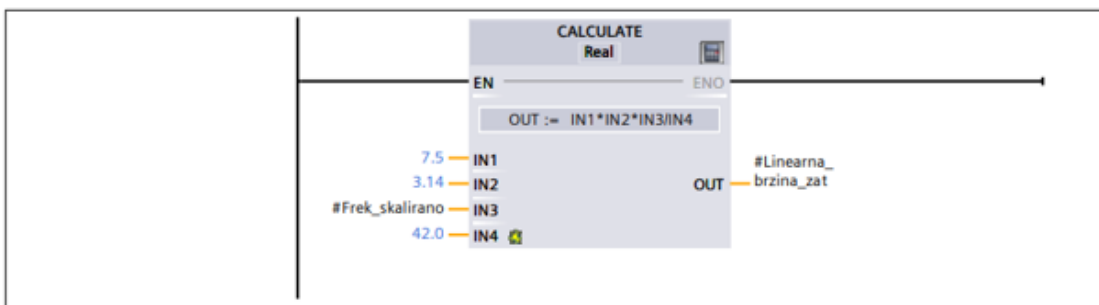
### Network 2:



### Network 3:



### Network 4:



Slika 6.18. Funkcijski blok „Frekvencija=>Linearna\_brzina\_zat“.

Ulazna analogna vrijednost je tipa *integer* koja ima raspon  $[-32768, 32767]$ , odnosno, u ovom slučaju  $[0, 32767]$ , budući da frekvencijski pretvarač može poslati samo pozitivne vrijednosti. Očitane je vrijednosti ponajprije potrebno skalirati u frekvencijski raspon  $[0, 50]$  da bismo ih mogli preračunati u linearnu brzinu.

Skaliranje najjednostavnije postignemo pomoću jednadžbe pravca kroz dvije točke, čiji izraz glasi:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1), \quad (6.13)$$

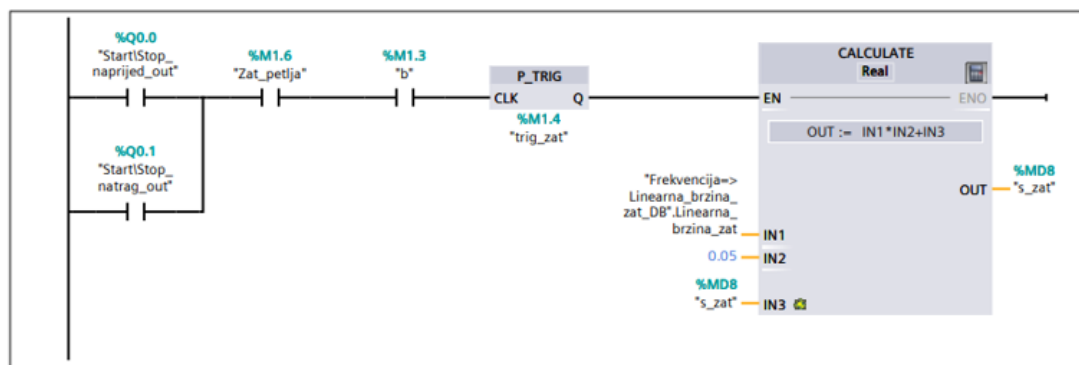
gdje je:  $x_1$  minimalna vrijednost analognog ulaza,  $x_2$  maksimalna vrijednost analognog ulaza,  $x$  trenutna očitana vrijednost analognog ulaza,  $y_1$  minimalna frekvencija,  $y_2$  maksimalna frekvencija,  $y$  trenutna frekvencija. Uvrstimo li poznate vrijednosti, dobijemo izraz:

$$y = \frac{50}{32767} \cdot x \quad (6.14)$$

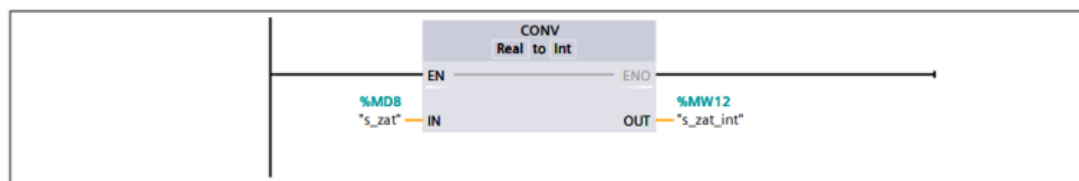
Dobiveni rezultat zatim ponovno dovodimo na blok koji preračunava frekvenciju u linearnu brzinu pokretne trake po jednadžbi (6.4).

Osim očitane analogne vrijednosti, ovaj FB kao ulaz također prima varijablu „smjer“ koja određuje dali je linearna brzina pozitivna ili negativna, to jest, dali se traka giba unaprijed ili unatrag.

Network 12:



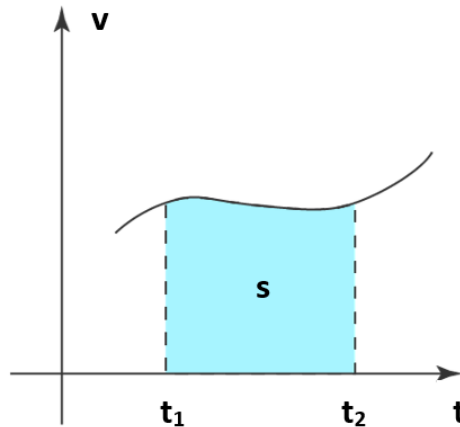
Network 13:



Slika 6.19. Network 12 i Network 13.

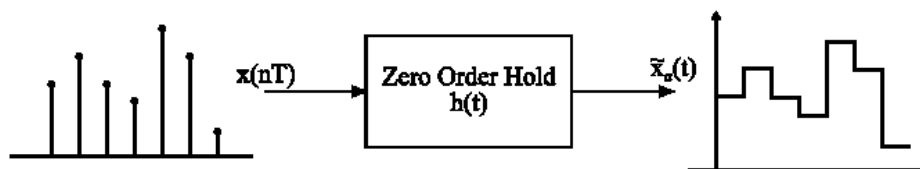


Pomoću izmjerene linearne brzine, moguće je odrediti trenutnu poziciju objekta na pokretnoj traci. Pozicija, odnosno, prijeđeni put jednak je površini ispod  $v-t$  krivulje (Slika 6.20.). Klasičan postupak određivanja površine ispod krivulje je računanjem integrala, međutim, analogni signal koji dobijemo na ulazu u PLC automatski se pretvara u digitalni. Da bismo izračunali površinu ispod krivulje, korištena je takozvana Zero Order Hold (ZOH) metoda.



Slika 6.20. Grafički prikaz odnosa  $v-t$  krivulje i prijeđenog puta.

Princip rada Zero Order Hold metode je takav da prvo uzrokujemo ulazni analogni signal, odnosno, očitamo vrijednosti signala u određenim vremenskim intervalima  $T$ . Očitana se vrijednost tada drži konstantnom sve do sljedećeg uzrokovanja, nakon čega se proces ponavlja. Na taj način dobijemo signal stepeničastog oblika (Slika 6.21.) čiju je površinu jednostavnije izračunati.



Slika 6.21. Rekonstrukcija signala ZOH metodom.

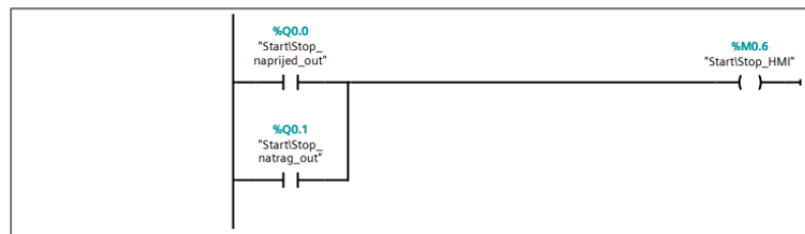
U ovom radu perioda uzrokovanja  $T$  iznosi 50 ms. Drugim riječima, prijeđeni put izračunamo tako da svakih 50 ms trenutnu linearnu brzinu pomnožimo s periodom  $T$ , te umnožak zbrojimo s prethodnim rezultatom, po izrazu:

$$s_n = v \cdot T + s_{n-1}, \quad (6.15)$$

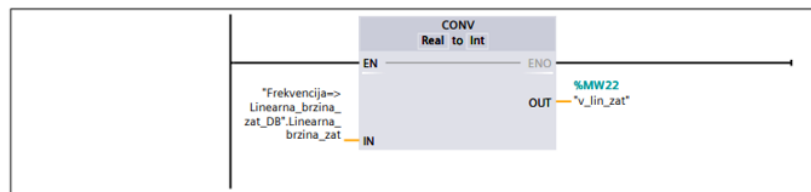
gdje je:  $n$  broj iteracije,  $s$  prijeđeni put,  $v$  linearna brzina,  $T$  perioda uzrokovanja.

Dobiveni rezultat ponovno dovodimo na *Network 14* i *Network 15* (Slika 6.14.), gdje se uspoređuje sa željenom pozicijom. Ako je trenutna pozicija manja od željene, program će aktivirati PLC izlaz  $\%Q0.0$ , te će se traka gibati unaprijed sve dok ne dođe do željene pozicije, nakon čega se zaustavlja. U suprotnom, ako je trenutna pozicija veća od željene, aktivira se izlaz  $\%Q0.1$ , te se traka počinje gibati unatrag do željene pozicije.

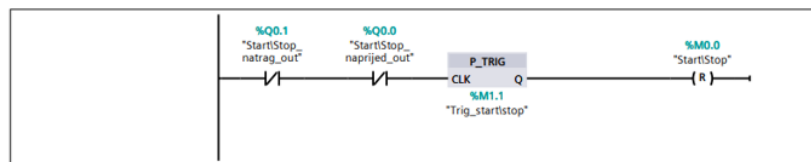
**Network 3:**



**Network 11:**



**Network 16:**

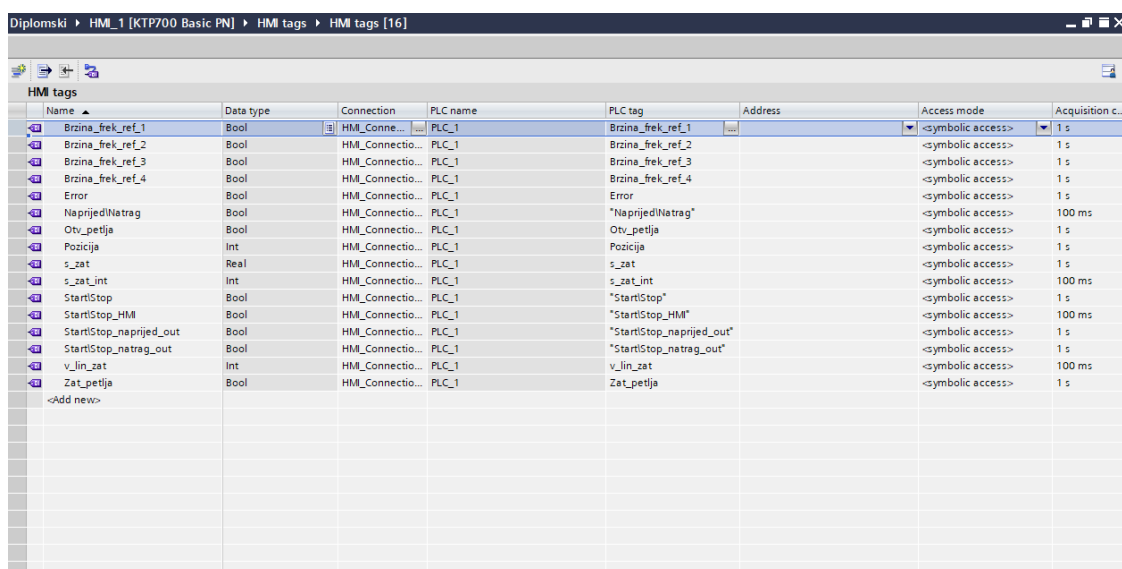


*Slika 6.22. Network 3, Network 11 i Network 16.*

*Network 3, 11 i 15* služe za upravljanje HMI varijablama, čiji rad će biti opisan u sljedećem poglavlju.

### 6.3. HMI vizualizacija

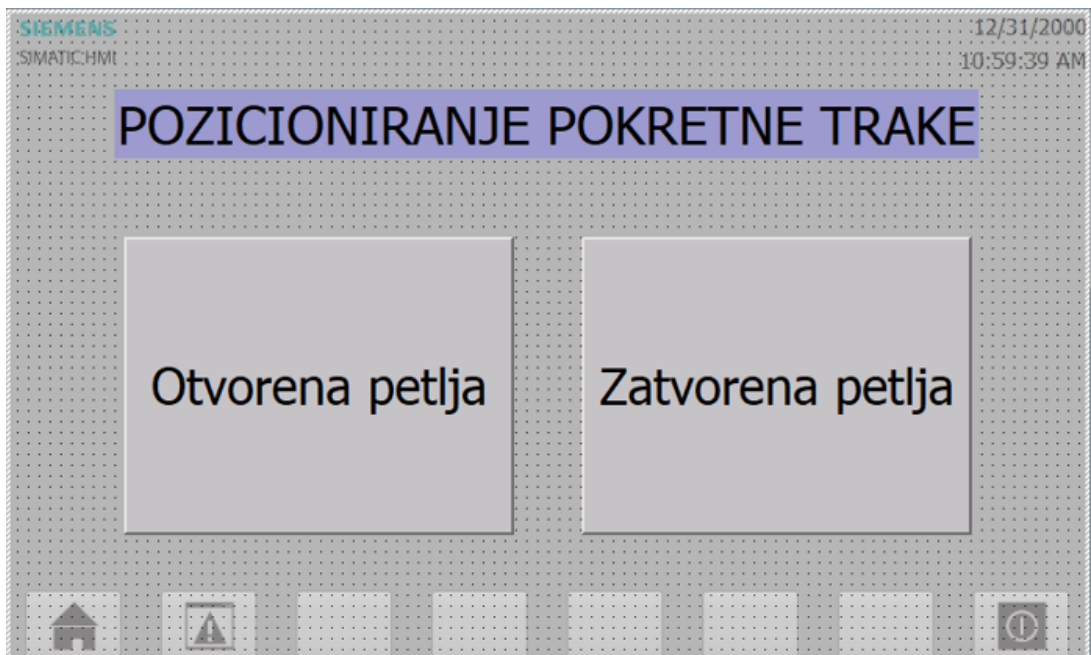
Nakon razvoja programskog koda, potrebno je izraditi HMI sučelje za razmjenu informacija između korisnika i sustava. Glavni cilj sučelja je omogućiti korisniku kontrolu i nadzor nad samim sustavom. Kako bi se uspostavila veza između HMI sučelja i samog programskog koda, potrebno je definirati HMI tagove koji će se koristiti u HMI sučelju. Definirane je tagove zatim potrebno povezati s tagovima programskog koda. To činimo tako da za svaki definirani tag, pod stupcem „PLC tag“ odaberemo tag programskog koda za koji želimo da definirani HMI tag bude vezan.



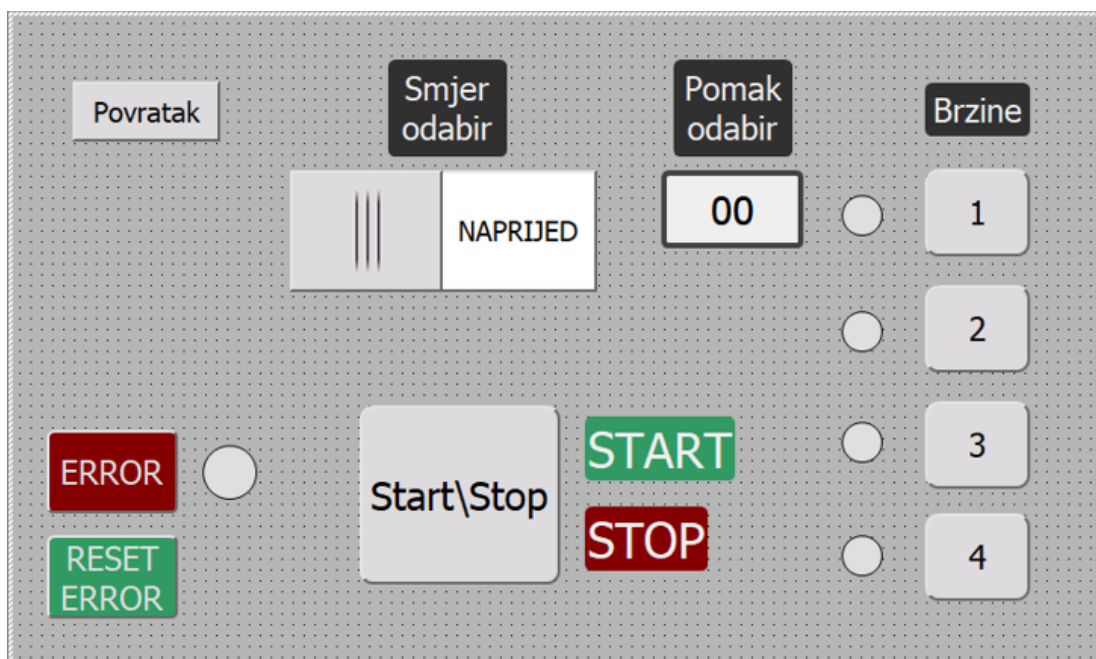
Name	Data type	Connection	PLC name	PLC tag	Address	Access mode	Acquisition c...
Brzina_frek_ref_1	Bool	HMI_Conne...	PLC_1	Brzina_frek_ref_1		<symbolic access>	1 s
Brzina_frek_ref_2	Bool	HMI_Conne...	PLC_1	Brzina_frek_ref_2		<symbolic access>	1 s
Brzina_frek_ref_3	Bool	HMI_Conne...	PLC_1	Brzina_frek_ref_3		<symbolic access>	1 s
Brzina_frek_ref_4	Bool	HMI_Conne...	PLC_1	Brzina_frek_ref_4		<symbolic access>	1 s
Error	Bool	HMI_Conne...	PLC_1	Error		<symbolic access>	1 s
NaprijedINatrag	Bool	HMI_Conne...	PLC_1	"NaprijedINatrag"		<symbolic access>	100 ms
Otv_petlja	Bool	HMI_Conne...	PLC_1	Otv_petlja		<symbolic access>	1 s
Pozicija	Int	HMI_Conne...	PLC_1	Pozicija		<symbolic access>	1 s
s_zat	Real	HMI_Conne...	PLC_1	s_zat		<symbolic access>	1 s
s_zat_int	Int	HMI_Conne...	PLC_1	s_zat_int		<symbolic access>	100 ms
StartIStop	Bool	HMI_Conne...	PLC_1	"StartIStop"		<symbolic access>	1 s
StartIStop_HMI	Bool	HMI_Conne...	PLC_1	"StartIStop_HMI"		<symbolic access>	100 ms
StartIStop_naprijed_out	Bool	HMI_Conne...	PLC_1	"StartIStop_naprijed_out"		<symbolic access>	1 s
StartIStop_natrag_out	Bool	HMI_Conne...	PLC_1	"StartIStop_natrag_out"		<symbolic access>	1 s
v_lin_zat	Int	HMI_Conne...	PLC_1	v_lin_zat		<symbolic access>	100 ms
Zat_petlja	Bool	HMI_Conne...	PLC_1	Zat_petlja		<symbolic access>	1 s
<Add new>							

Slika 6.23. Definirani HMI tagovi.

Nakon definiranja HMI tagova, slijedi izrada same vizualizacije. Izrađeno je 5 različitih zaslona: početni izborni zaslon, zaslon za upravljanje u otvorenoj petlji, zaslon za upravljanje u zatvorenoj petlji, zaslon s trendom pozicije, te zaslon s trendom brzine. U nastavku su prikazani izrađeni zaslono, te je nadalje dan detaljan opis svakog zaslona.



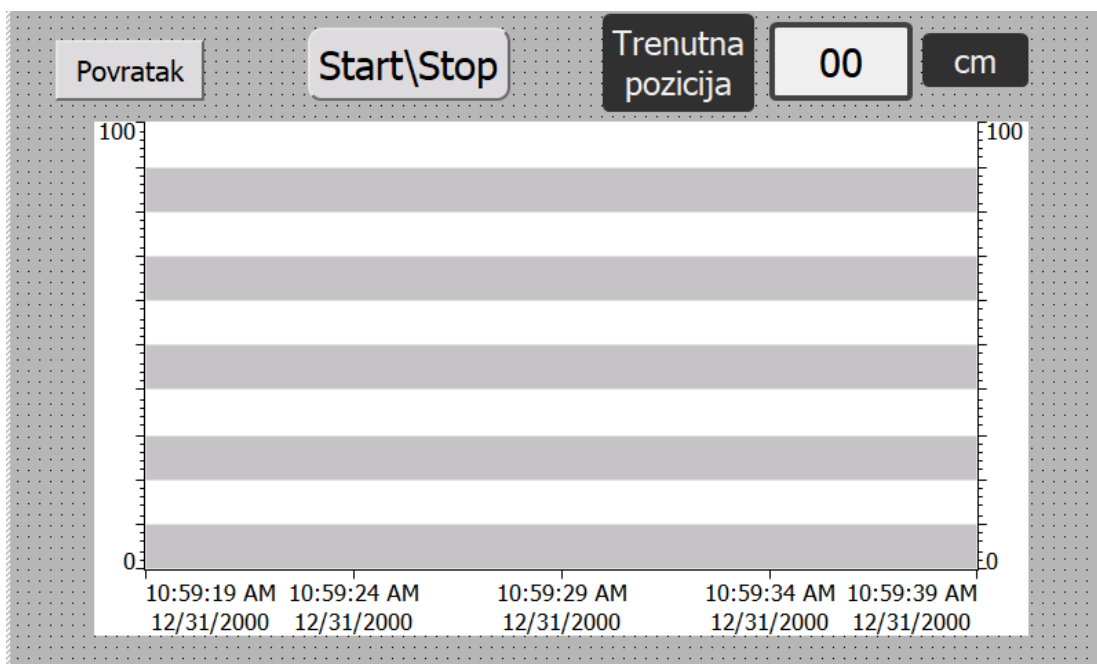
Slika 6.24. Početni izborni zaslon.



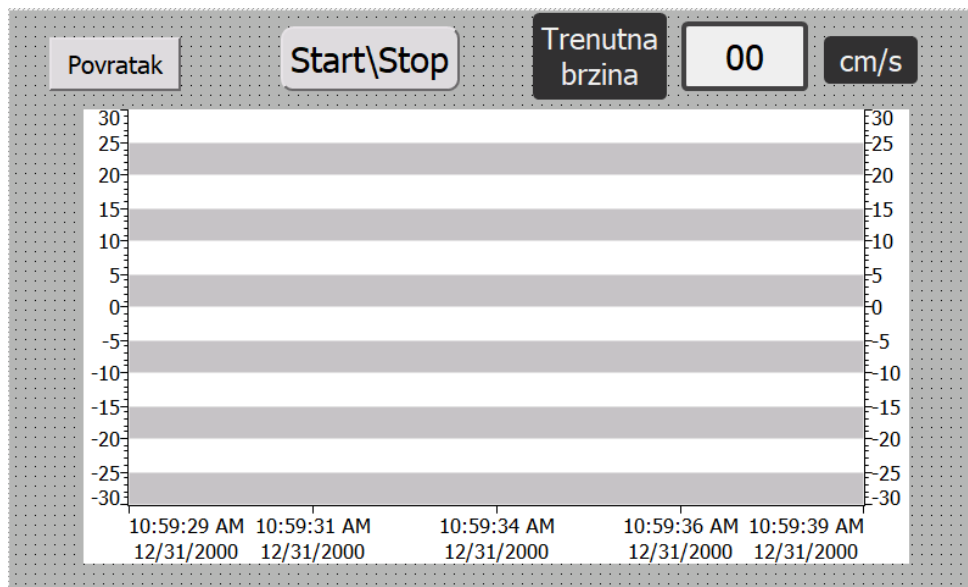
Slika 6.25. Zaslon za upravljanje u otvorenoj petlji.



Slika 6.26. Zaslona za upravljanje u zatvorenoj petlji.



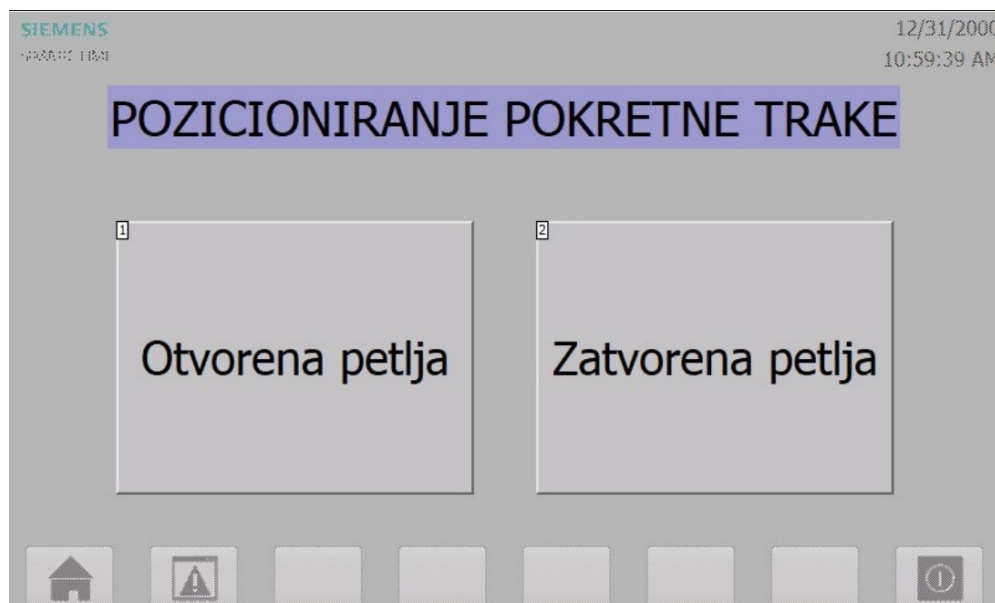
Slika 6.27. Zaslona s trendom pozicije.



Slika 6.28. Zaslonski prikaz trenda brzine.

### 6.3.1. Početni izborni zaslon

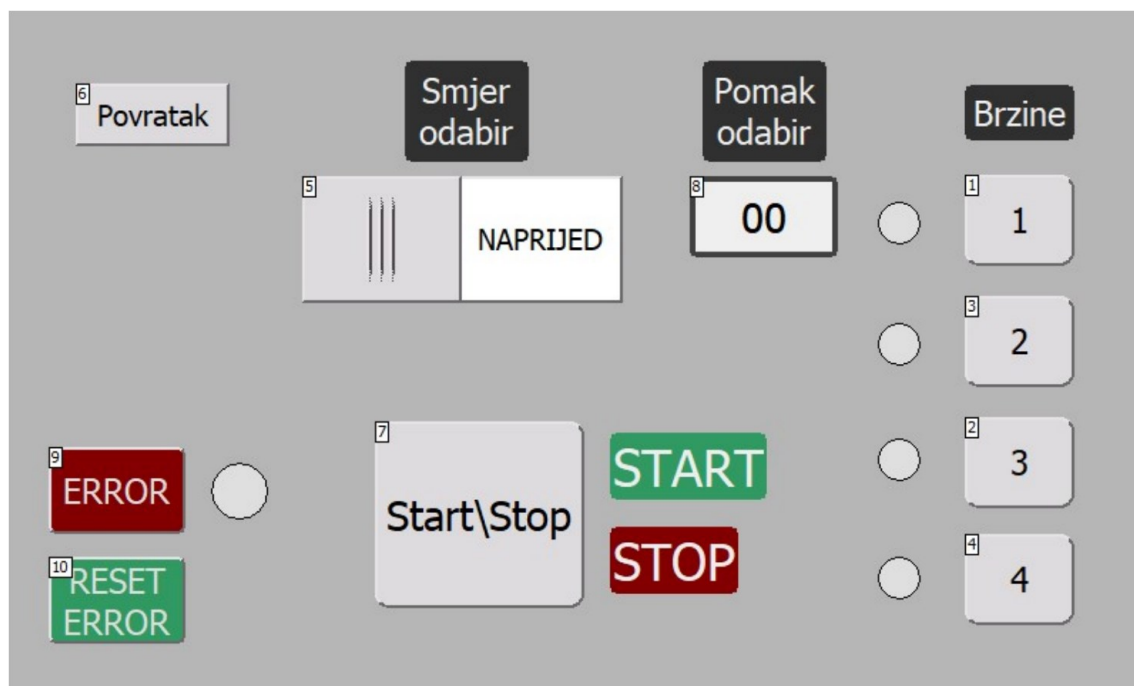
Na početnom izbornom zaslonu imamo mogućnost odabira želimo li upravljati pokretnom trakom u otvorenoj ili zatvorenoj petlji. Pritiskom na jednu od opcija će nas odvesti na novi, za to odgovarajući zaslon.



Slika 6.29. Početni zaslon s označenim dijelovima: 1) Odabir upravljanja trakom u otvorenoj petlji, 2) Odabir upravljanja trakom u zatvorenoj petlji.

### 6.3.2. Zaslona za upravljanje u otvorenoj petlji

Sljedeći zaslon je zaslon za pozicioniranje trake u otvorenoj petlji, čije označene dijelove možemo vidjeti na Slici 6.30.



Slika 6.30. Zaslona za upravljanje u otvorenoj petlji s označenim dijelovima: 1)-4) Brzine, 5) Odabir smjera, 6) Povratak na početni zaslon, 7) Pokretanje/zaustavljanje trake, 8) Odabir pomaka, 9) Signalizacija greške, 10) Resetiranje greške.

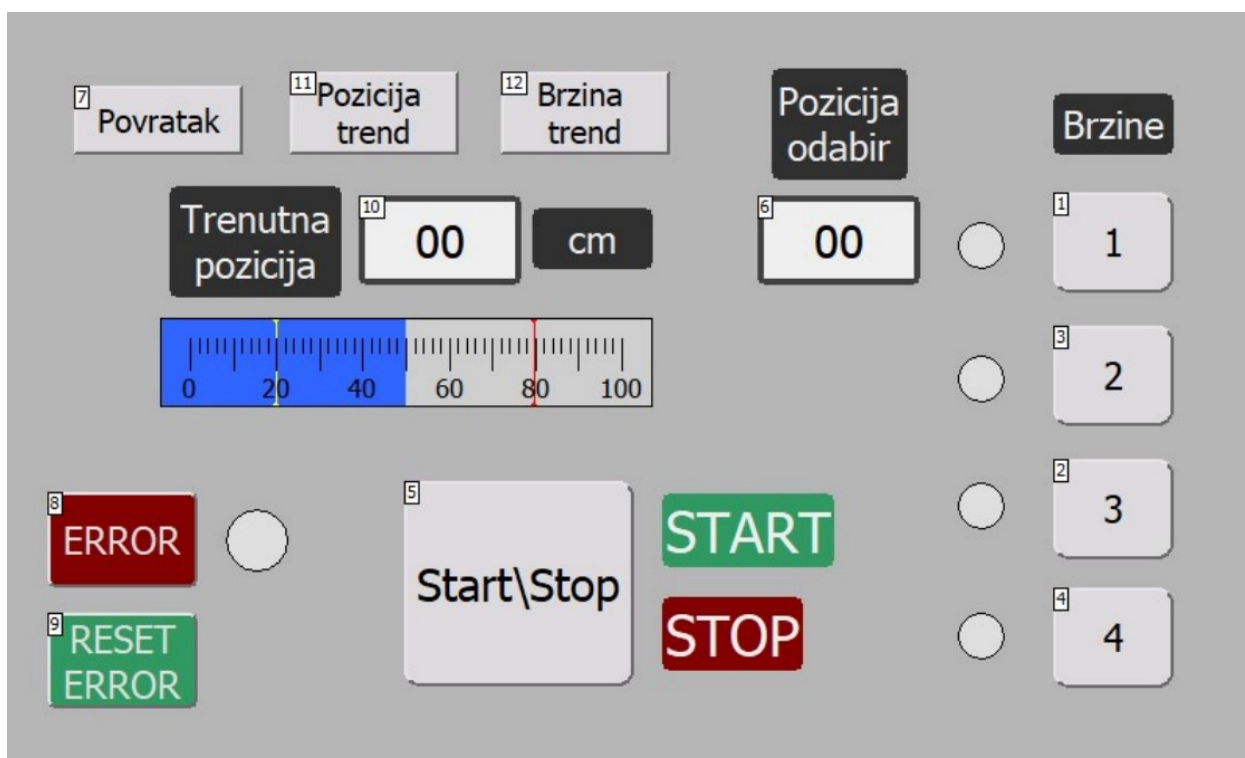
Da bismo upravljali trakom u otvorenoj petlji, potrebno je odabrati željenu brzinu, pomak i smjer gibanja trake. Kao što je već navedeno, na raspolaganju imamo četiri različite brzine koje odaberemo pritiskom na jedno od tipkala označeno brojevima 1-4, nakon čega se uključuje lampica pored odabrane brzine kao indikator. Zatim odaberemo pomak, odnosno, koliki put želimo da traka prođe. Sama pokretna traka duža je malo više od 1 metar, te je korisniku omogućeno da može pomicati predmet na traci do 1 metar tako da upiše vrijednost u rasponu 0-99, gdje vrijednosti označavaju centimetre. Na kraju, pritiskom na prekidač *NAPRIJED/NATRAG* korisnik bira željeni smjer gibanja trake. Nakon odabira svih vrijednosti, traka se može pokrenuti pritiskom na tipkalo *Start/Stop*. Lampice *START* i *STOP* pored tipkala služe za indikaciju dali je traka u pogonu ili miruje. Odabrane vrijednosti nije moguće mijenjati ako je traka u pogonu. Drugim riječima, ako

želimo promijeniti neku od vrijednosti, potrebno je traku ručno zaustaviti pritiskom na tipkalo ili pričekati da dođe do željene pozicije.

Osim navedenih dijelova, zaslon također sadrži tipkala za signalizaciju i resetiranje greške (Error), te tipkalo za povratak na početni izborni zaslon. Pritiskom na tipkalo za signalizaciju greške, traka se zaustavlja, te ju nije moguće ponovno pokrenuti sve dok ne pritisnemo tipkalo za resetiranje greške.

### 6.3.3. Zaslon za upravljanje u zatvorenoj petlji

Sljedeće imamo zaslon za pozicioniranje trake u zatvorenoj petlji, čije označene dijelove možemo vidjeti na Slici 6.31.



Slika 6.31. Zaslon za upravljanje u zatvorenoj petlji s označenim dijelovima: 1)-4) Brzine, 5) Pokretanje/zaustavljanje trake, 6) Odabir pozicije, 7) Povratak na početni zaslon, 8) Signalizacija greške, 9) Resetiranje greške, 10) Trenutna pozicija, 11) Prijelaz na zaslon s trendom pozicije, 12) Prijelaz na zaslon s trendom brzine.

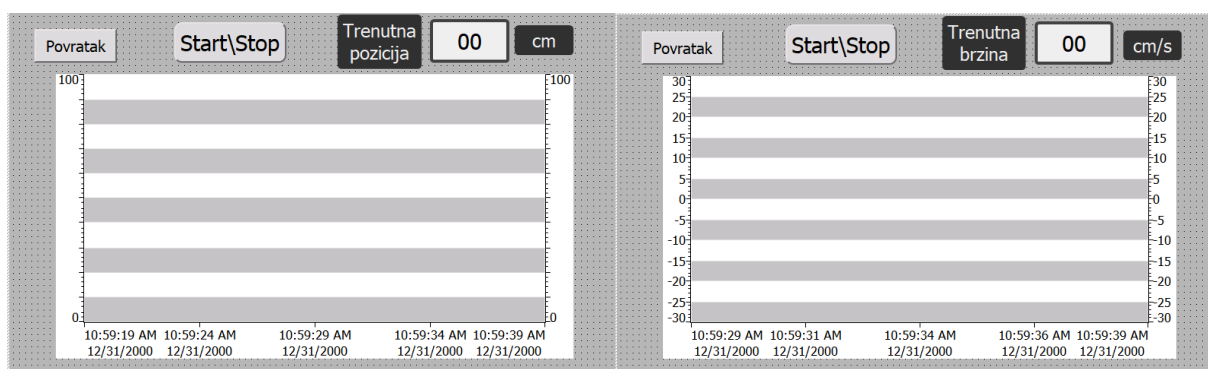


Princip rada zaslona kod zatvorene petlje sličan je kao i kod otvorene petlje, s jedinom razlikom to što je kod otvorene petlje korisnik trebao odabrati pomak i smjer, dok je kod zatvorene petlje potrebno odabrati samo poziciju. Pozicija se također bira unosom vrijednosti u rasponu 0-99. 0 označava jedan kraj trake, dok 99 označava drugi kraj, te program automatski izračuna u kojem je smjeru i za koju udaljenost potrebno pomaknuti traku. Zaslone u zatvorenoj petlji također sadrži dva pokazivača stvarne trenutne pozicije.

Osim navedenih dijelova, zaslon još sadrži dva tipkala pomoću kojih prelazimo na zaslone s trendovima, koji su opisani u nastavku.

Ostali dijelovi zaslona funkcioniraju identično kao i kod otvorene petlje.

#### 6.3.4. Zaslone s trendovima



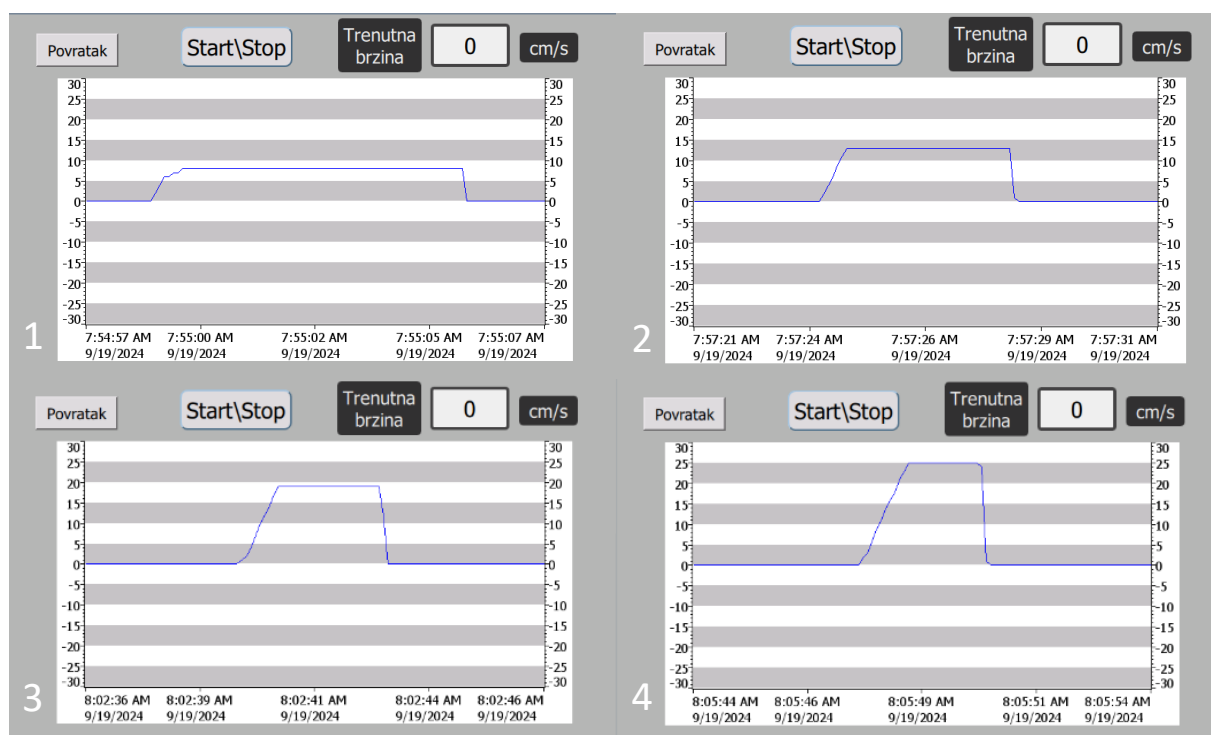
Slika 6.32. Zaslone s trendovima pozicije (lijevo) i brzine (desno).

Posljednja dva zaslona služe za grafički prikaz trenutne vrijednosti određene veličine kroz vrijeme. U našem slučaju, to su trenutna pozicija i trenutna brzina pokretne trake. Osim grafički, zaslone također prikazuju numeričku vrijednost spomenutih veličina. Pomoću navedenih zaslona možemo provesti testiranje izrađenog sustava, te dobivene rezultate usporediti međusobno i s očekivanim rezultatima. U daljnjem poglavlju je opisano takvo testiranje, odnosno, funkcionalno ispitivanje sustava.

## 7. FUNKCIONALNO ISPITIVANJE SUSTAVA

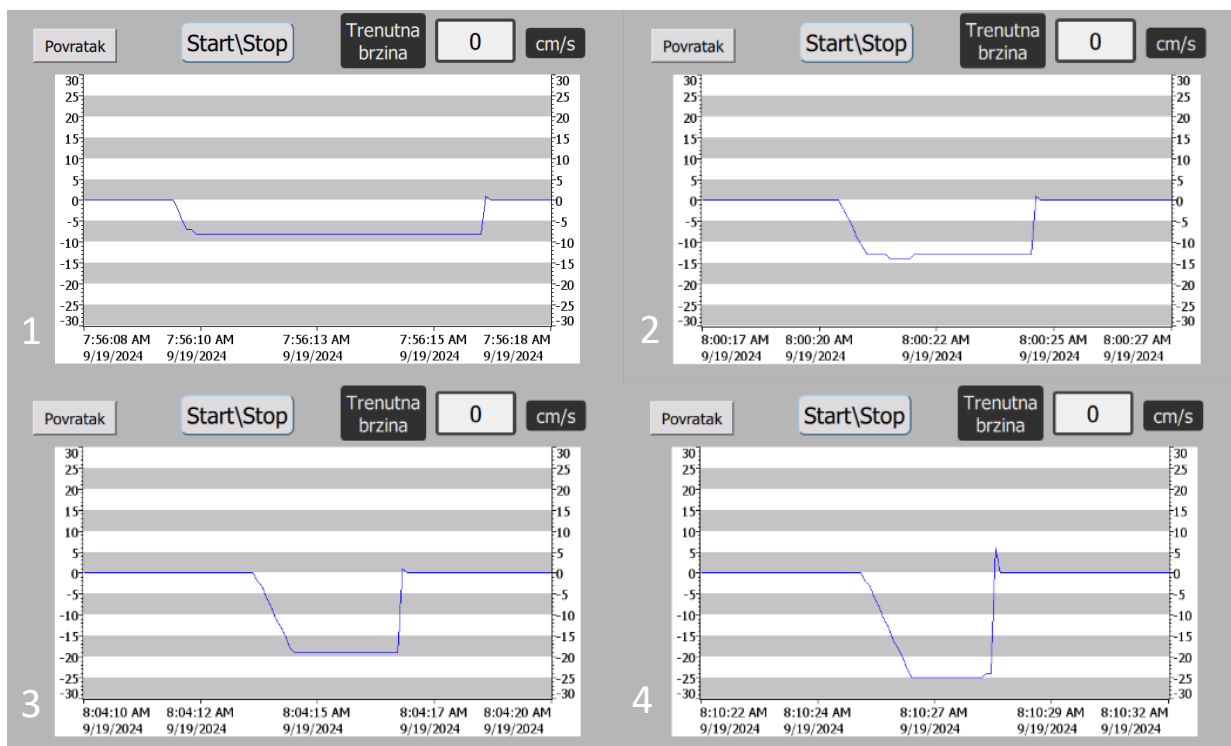
U svrhu funkcionalnog ispitivanja provedena su dva testiranja: testiranje odziva brzine i pozicije.

U nastavku su detaljnije opisana oba testiranja.



Slika 7.1. Odzivi brzina (1 – brzina 1, 2 – brzina 2, 3 – brzina 3, 4 – brzina 4).

Prvo testiranje uključuje odziv brzine. Za svaku od četiri brzina je postavljeno da traka prijeđe pola ukupnog puta (od vrijednosti 0 do 50), te se preko grafičkog prikaza u stvarnom vremenu motre izmjerene brzine trake, kao što je prikazano na Slici 7.1.. Na grafovima možemo uočiti da za svaku brzinu traka ima drugačije vrijeme gibanja, što je i očekivano budući da pri većim brzinama traka prije dođe do željene pozicije. Također možemo uočiti konstantno vrijeme ubrzanja za sve brzine (1 sekunda), te nagli pad brzine kada traka dođe do pozicije. Također je provedeno još jedno testiranje brzine u suprotnom smjeru (od vrijednosti 50 do 0) koje možemo vidjeti u nastavku na Slici 7.2..



Slika 7.2. Odzivi brzina u suprotnom smjeru (1–brzina 1, 2–brzina 2, 3 – brzina 3, 4 – brzina 4).

Ponovno, na grafovima možemo uočiti da su odzivi brzina identični onima na Slici 7.1., samo što su vrijednosti negativne budući da se traka giba u suprotnom smjeru.



Slika 7.3. Odzivi pozicija (1–brzina 1, 2–brzina 2, 3 – brzina 3, 4 – brzina 4).

Drugo testiranje se odnosi na praćenje trenutne pozicije kroz vrijeme. Kao i u testiranju brzine, postavljeno je da traka prijeđe pola ukupnog puta za svaku od četiri brzina, te se na grafičkom prikazu u stvarnom vremenu promatra promjena pozicije. Ponovno uočavamo da povećanjem brzine traka dođe do željene pozicije u kraćem vremenu.

## 8. ZAKLJUČAK

Automatizacija, uključujući pokretne trake, postala je temeljni element moderne industrije, oblikujući nove standarde proizvodnje. Automatizirani sustavi ne samo da donose veću brzinu i preciznost pri proizvodnji, nego omogućuju i fleksibilnost u prilagodbi promjenama u potražnji.

Cilj ovog diplomskog rada bio je projektirati sustav i razviti upravljački program za pozicioniranje predmeta na pokretnoj traci u otvorenoj i zatvorenoj petlji. U radu su detaljno opisane sve korištene komponente sustava, poput asinkronog motora, frekvencijskog pretvarača, PLC-a itd., te objašnjen njihov međusobni odnos. Također je dan detaljan opis korištenih metoda za upravljanje i pozicioniranje, kao i opis same programske aplikacije koja upravlja cijelim sustavom. Korisniku je omogućeno upravljanje sustavom preko HMI-a odabirom brzine, te pomaka ili pozicije.

Ovaj rad je nadalje moguće dodatno poboljšati implementiranjem dodatnih komponenti poput enkodera za preciznije mjerenje brzine, ili senzora blizine za detektiranje predmeta na pokretnoj traci.

## LITERATURA

- [1] Matika, D.: „Automatsko upravljanje – prvo predavanje“, predavanja iz kolegija „Automatsko upravljanje“, Sveučilište u Rijeci – Tehnički fakultet, 2021.
- [2] „Open Loop Control System“, s Interneta, <https://www.geeksforgeeks.org/open-loop-control-system/>, 24. svibnja 2024.
- [3] „Closed-Loop Control System“, s Interneta, <https://www.geeksforgeeks.org/closed-loop-control-system/>, 27. veljače 2024.
- [4] Prenc, R.: „Asinkroni Strojevi“, predavanja iz kolegija „Električni strojevi“, Sveučilište u Rijeci - Tehnički fakultet, 2021.
- [5] Bulić, N.: „Asinkroni stroj“, predavanja iz kolegija „Elektromotorni pogoni“, Sveučilište u Rijeci – Tehnički fakultet, 2011.
- [6] „Programmable logic controller“, s Interneta, [https://en.wikipedia.org/wiki/Programmable\\_logic\\_controller](https://en.wikipedia.org/wiki/Programmable_logic_controller), 15. svibnja 2024.
- [7] „What is a Programmable Logic Controller“, s Interneta, <https://www.geeksforgeeks.org/what-is-a-programmable-logic-controllers-plc/>, 27. veljače 2024.
- [8] „Siemens Industry Mall“, s Interneta, <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7214-1AG40-0XB0>
- [9] Bulić, N.: „Učinski pretvarači za EMP s asinkronim strojem“, predavanja iz kolegija „Upravljanje elektromotornim pogonima“, Sveučilište u Rijeci – Tehnički fakultet, 2011.
- [10] „Variable-frequency drive“, s Interneta, [https://en.wikipedia.org/wiki/Variable-frequency\\_drive](https://en.wikipedia.org/wiki/Variable-frequency_drive), 2. veljače 2024.
- [11] „Yaskawa“, s Interneta, [https://www.no.yaskawa.eu.com/products/ac-drives/legacy-drives/seriesdetail/serie/j1000-series\\_516](https://www.no.yaskawa.eu.com/products/ac-drives/legacy-drives/seriesdetail/serie/j1000-series_516)
- [12] Yaskawa: „YASKAWA AC Drive-J1000 Compact V/f Control Drive Technical Manual“, 2008.
- [13] „Reduktor“, s Interneta, <https://hr.wikipedia.org/wiki/Reduktor>, 20. prosinac 2021.
- [14] „Industrial Belt Conveyors“, s Interneta, <https://www.ergonomicpartners.com/lewco-industrial-belt-conveyors-belt-roller-poly-pbr>

- [15] „Totally Integrated Automation Portal – Always ready for tomorrow“, s Interneta, [https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html?\\_originalHost=new.siemens.com](https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html?_originalHost=new.siemens.com), 22. kolovoza 2024.
- [16] „Siemens PLC Programming Blocks | How to use it?“, s Interneta, <https://instrumentationblog.com/siemens-plc-programming-blocks/>, 23. kolovoza 2024.
- [17] Volarić, I.: „Uvod u PLC“, predavanja iz kolegija „Elementi automatizacije postrojenja“, 2021.
- [18] „Uvodna laboratorijska vježba“, predavanja iz kolegija „Automatizacija postrojenja i procesa“

## SAŽETAK I KLJUČNE RIJEČI

U ovome je diplomskom radu obrađena tema implementacije pozicioniranja pokretne trake. Izrađen je sustav koji omogućuje korisniku da pozicionira predmet na stvarnom modelu pokretne trake u otvorenoj i zatvorenoj petlji. Dan je detaljan opis sustava, njegovih komponenti, te programske aplikacije. Programska aplikacija je izrađena u programskom okruženju TIA Portal. Program je zatim implementiran na PLC S7-1200 koji upravlja pozicijom pokretne trake posredstvom frekvencijskog pretvarača Yaskawa i asinkronog motora.

**Ključne riječi:** automatizacija, upravljanje, PLC, TIA Portal, frekvencijski pretvarač, asinkroni motor



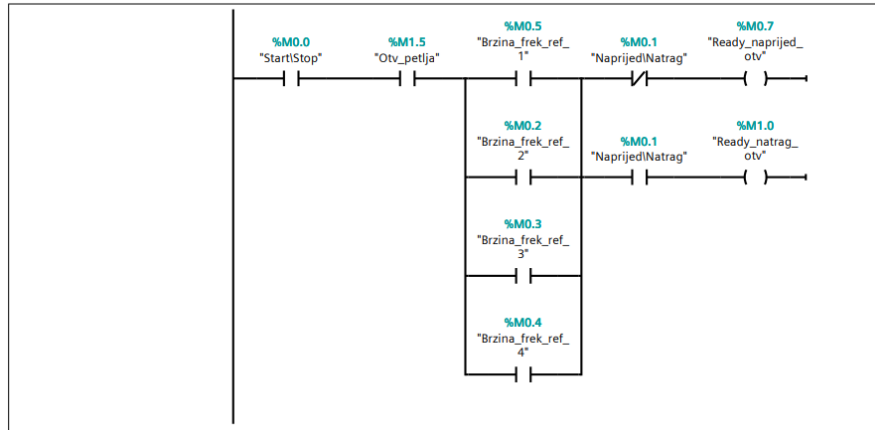
## **SUMMARY AND KEYWORDS**

This master's thesis discusses the topic of conveyor belt positioning implementation. A system has been developed that enables the user to position an object on a real-life conveyor belt model in both open and closed loop control systems. A detailed description of the system, its components, and the software application is provided. The software application was developed in the TIA Portal programming environment. The program was then implemented on a PLC S7-1200, which controls the position of the conveyor belt through a Yaskawa variable-frequency drive and an induction motor.

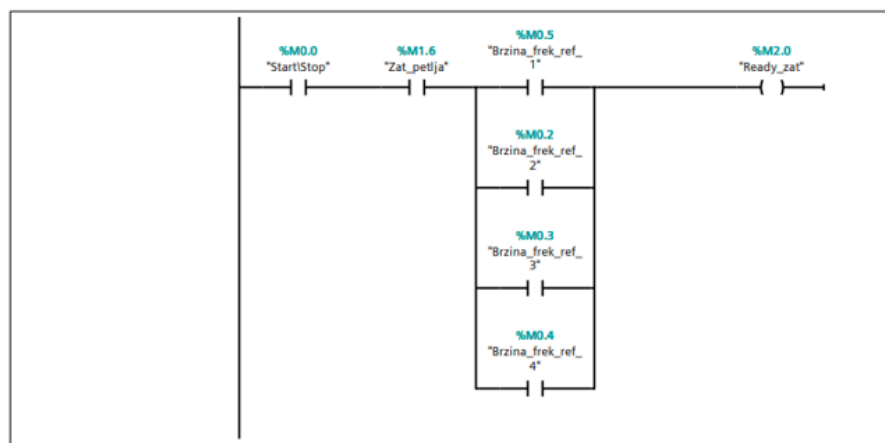
**Keywords:** automation, system control, PLC, TIA Portal, variable-frequency drive, induction motor

# DODATAK A: PROGRAMSKI KOD

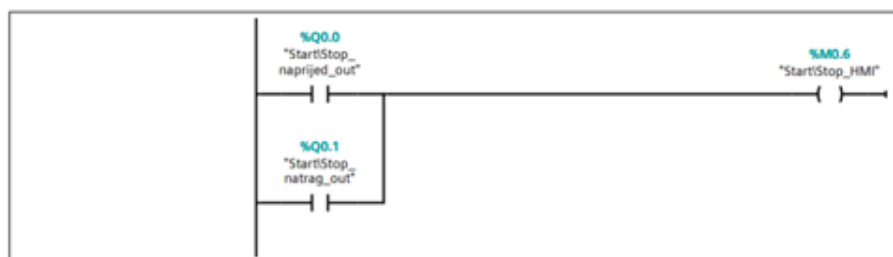
Network 1:



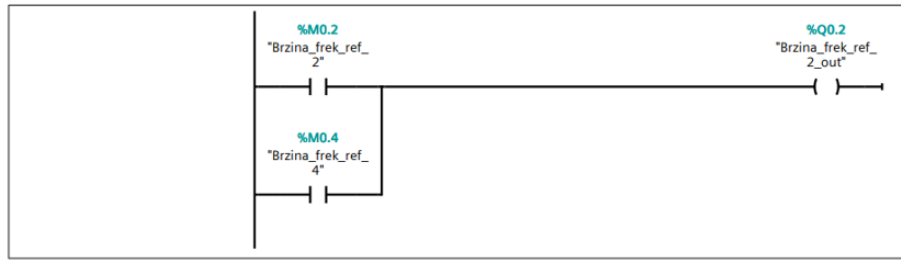
Network 2:



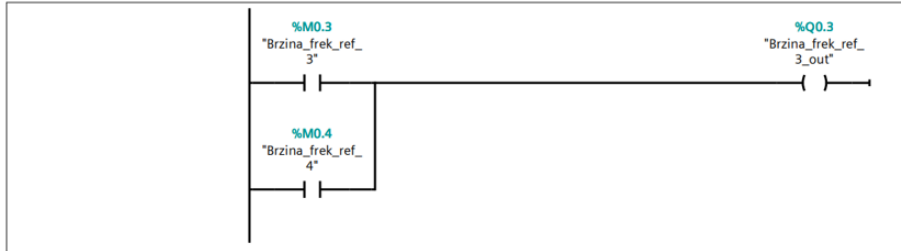
Network 3:



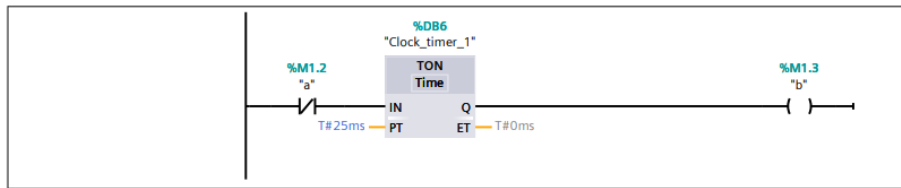
**Network 4:**



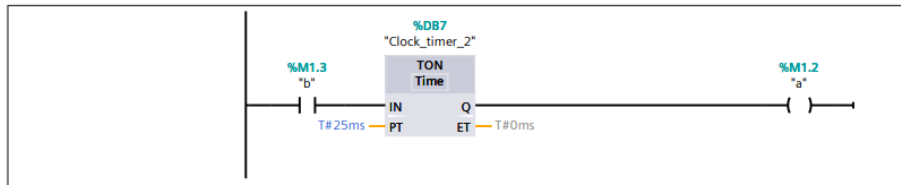
**Network 5:**



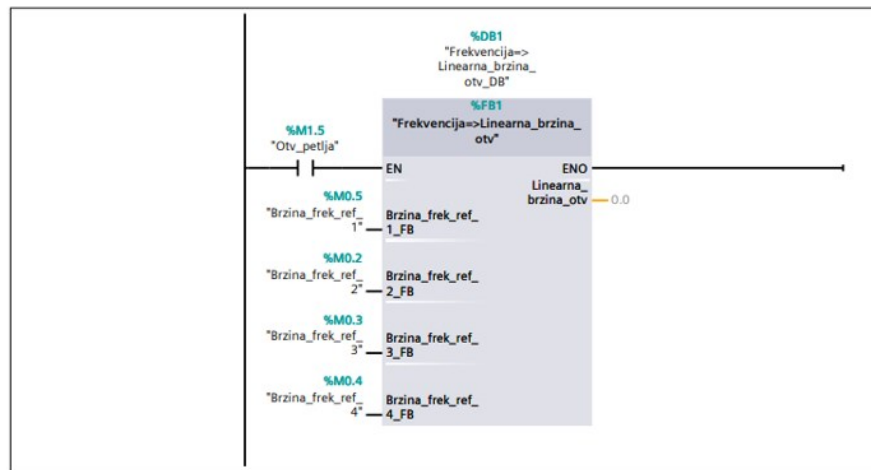
**Network 6:**



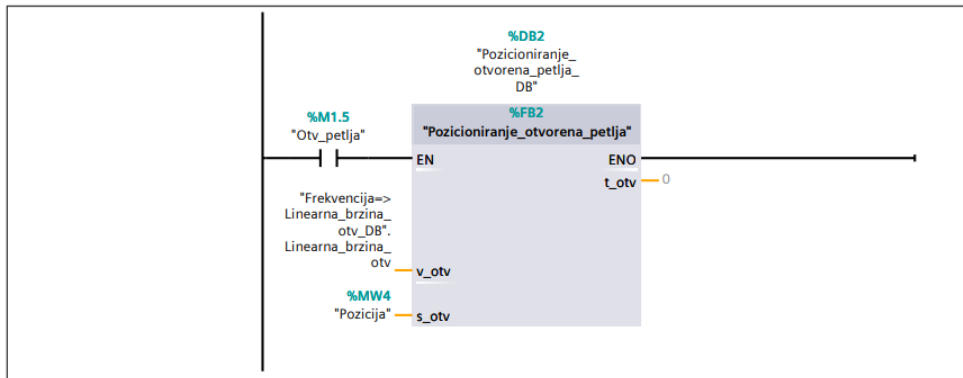
**Network 7:**



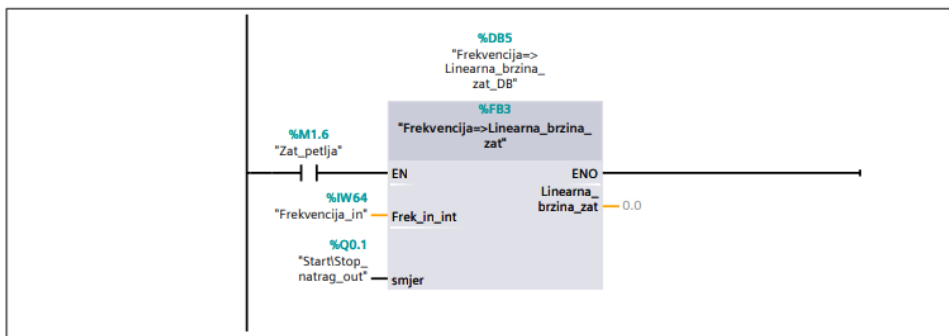
**Network 8:**



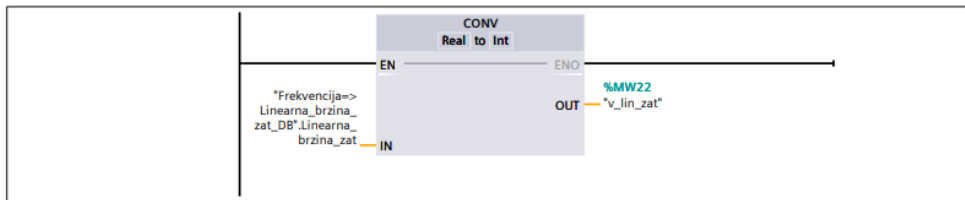
**Network 9:**



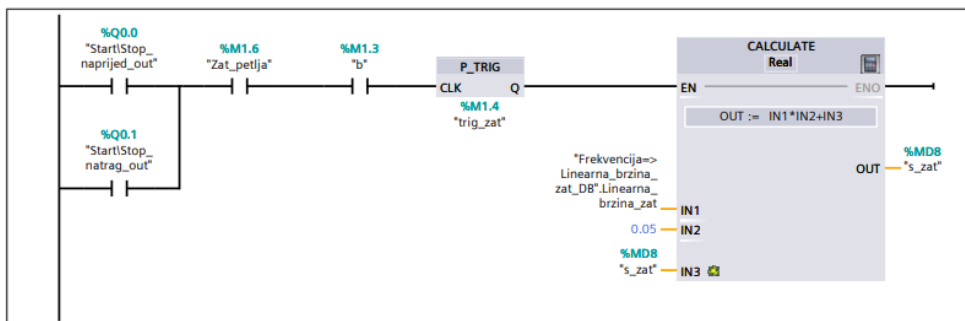
**Network 10:**



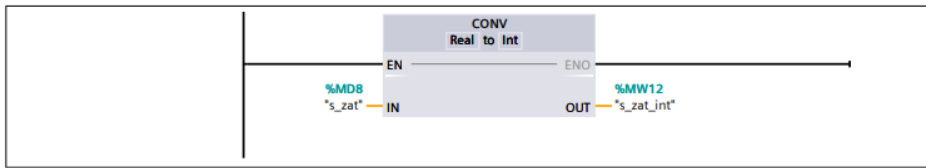
**Network 11:**



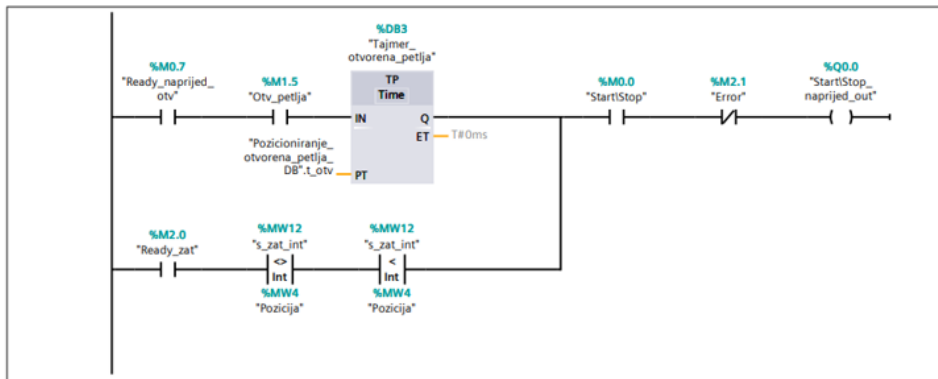
**Network 12:**



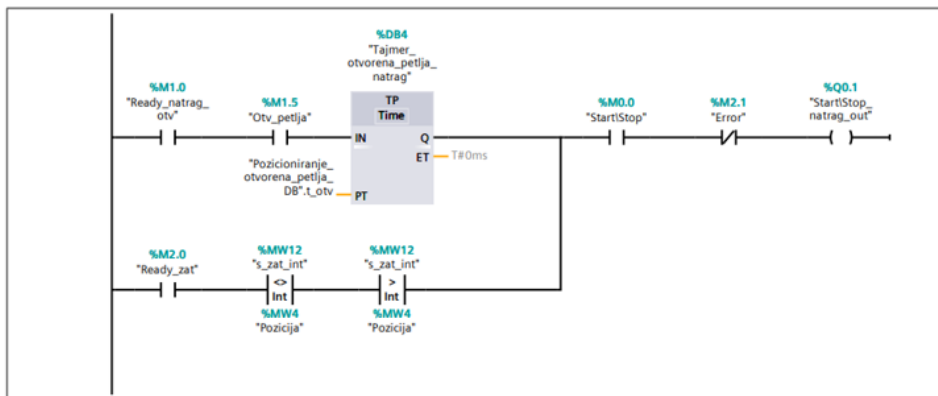
Network 13:



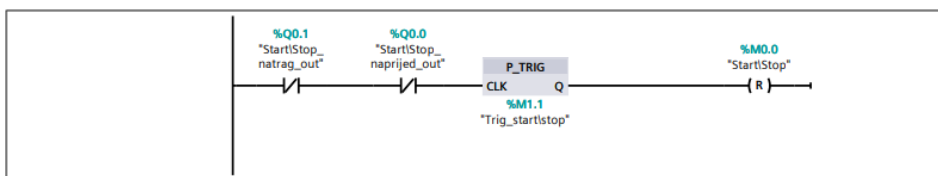
Network 14:



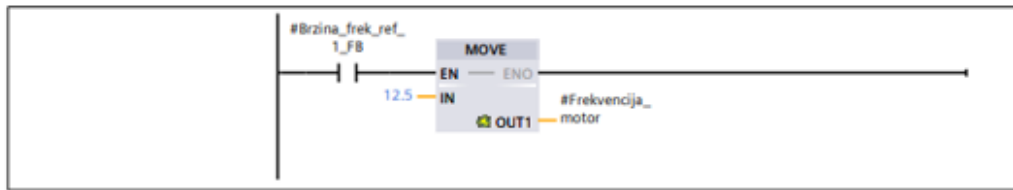
Network 15:



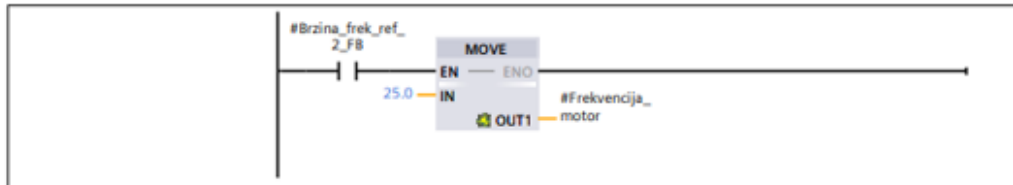
Network 16:



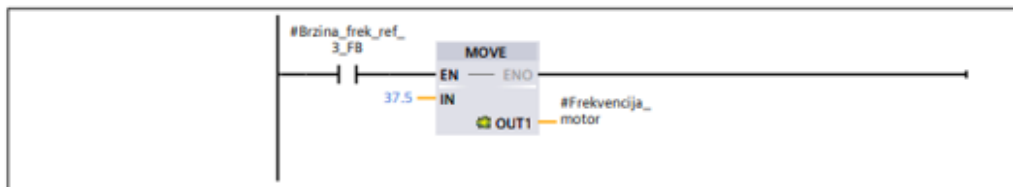
### Network 1:



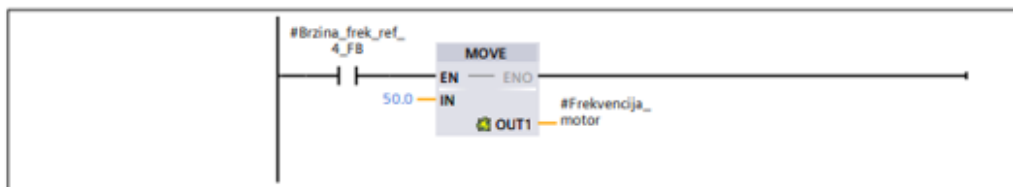
### Network 2:



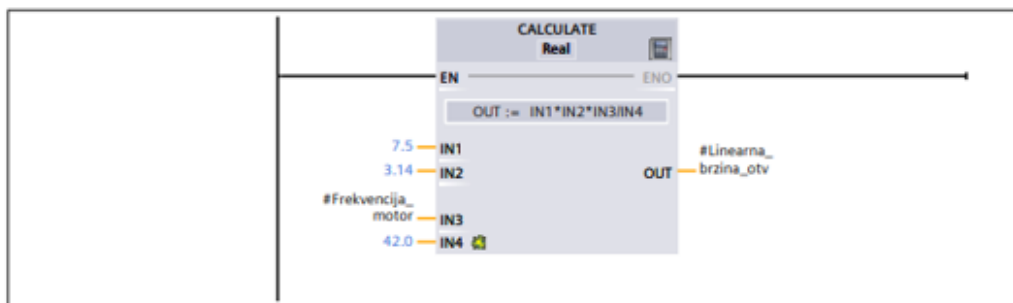
### Network 3:



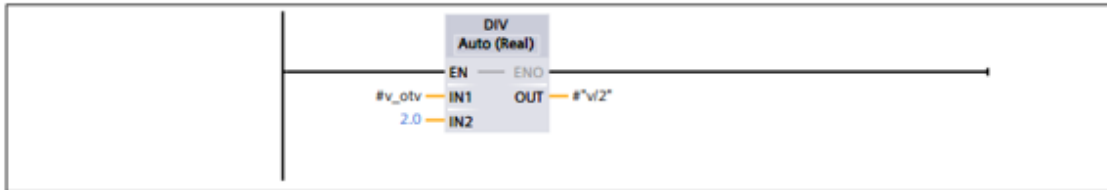
### Network 4:



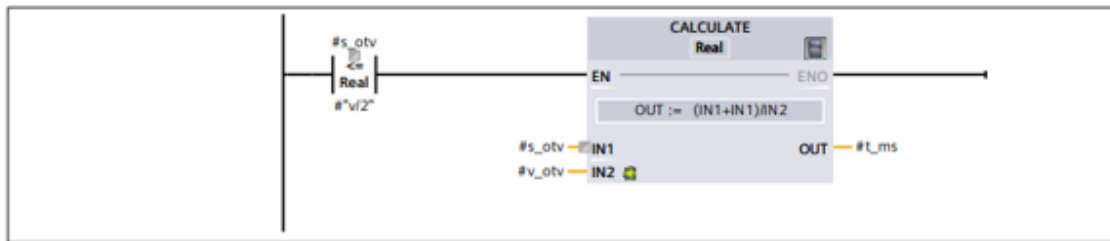
### Network 5:



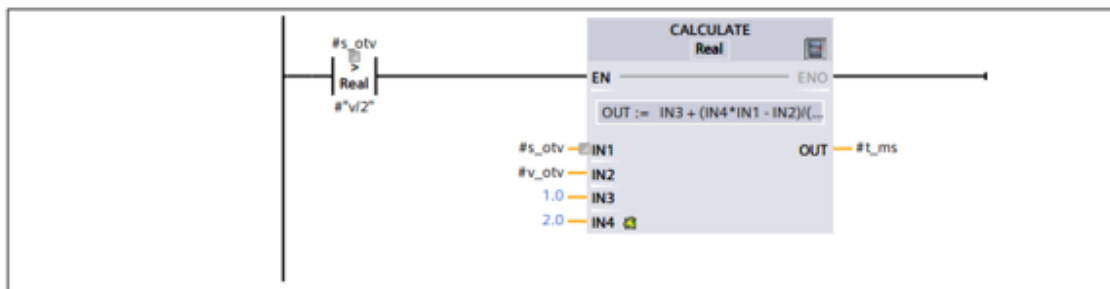
**Network 1:**



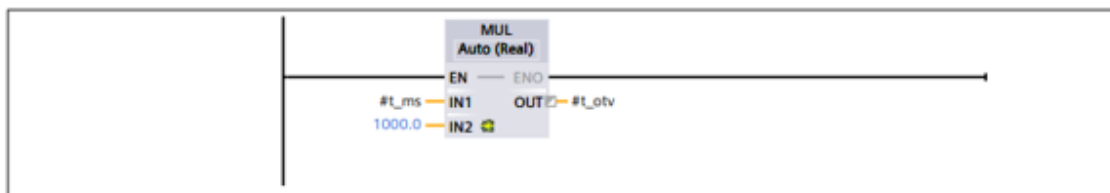
**Network 2:**



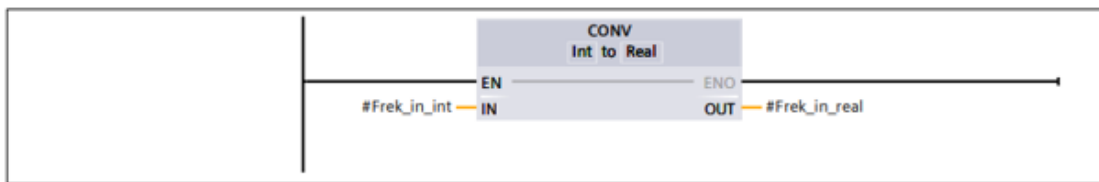
**Network 3:**



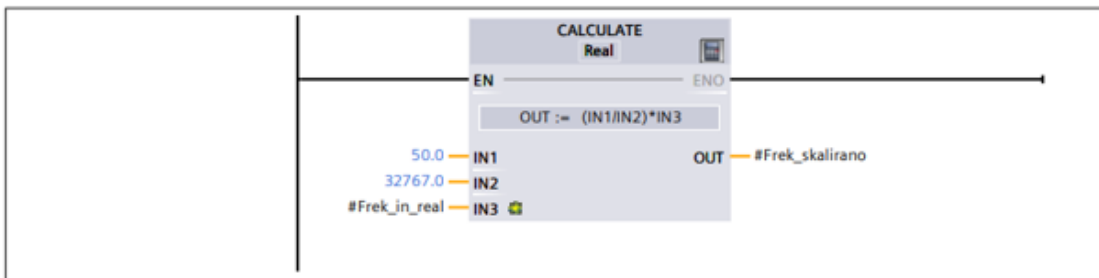
**Network 4:**



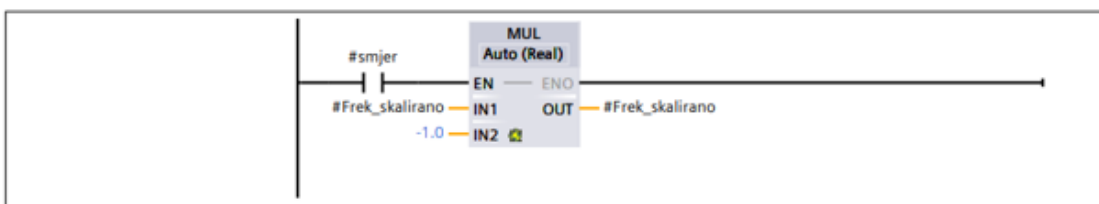
### Network 1:



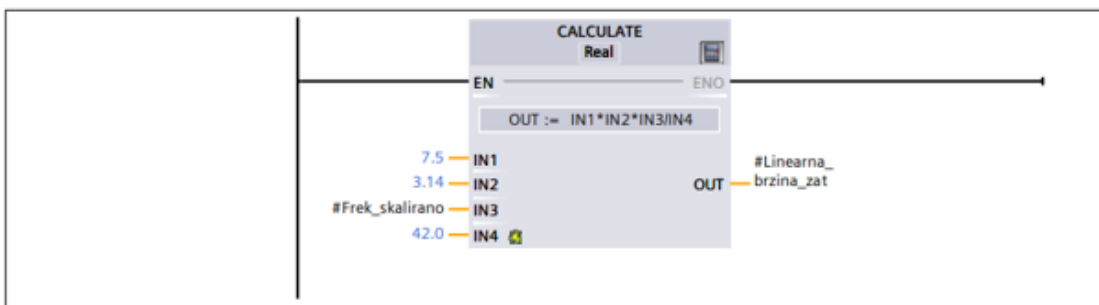
### Network 2:



### Network 3:



### Network 4:





# POPIS SLIKA I TABLICA

## Popis slika

Slika 2.1. Shema upravljačkog sustava u otvorenoj petlji.....	3
Slika 2.2. Shema upravljačko sustava u zatvorenoj petlji.....	4
Slika 3.1. Kavezni asinkroni motor s označenim dijelovima.....	5
Slika 3.2. Trofazna struja u statoru motora.....	6
Slika 3.3. Karakteristika skalarnog upravljanja.....	9
Slika 3.4. SIMATIC S7-1200 CPU 1214C DC/DC/DC modul.....	10
Slika 3.5. Ciklus PLC-a.....	10
Slika 3.6. HMI panel.....	10
Slika 3.7. Shema neizravnog frekvencijskog pretvarača.....	11
Slika 3.8. Yaskawa J1000 frekvencijski pretvarač.....	13
Slika 3.9. Shema priključaka u energetsom krugu.....	13
Slika 3.10. Shema priključaka u upravljačkom krugu.....	14
Slika 3.11. Upravljanje brzinom vrtnje putem upravljačkog kruga.....	15
Slika 3.13. Pokretna traka.....	18
Slika 3.12. Reduktor spojen na asinkroni motor.....	18
Slika 4.1. Dijagram ožičenja upravljačkog sustava.....	19
Slika 4.2. Sustav upravljanja s označenim komponentama(1).....	20
Slika 4.3. Sustav upravljanja s označenim komponentama(2).....	20
Slika 5.1. TIA Portal početni zaslon.....	22
Slika 5.2. Odabir programskog bloka u TIA Portalu.....	23
Slika 6.1. Izbornik za dodavanje uređaja.....	25
Slika 6.2. Promjena IP adrese IPv4.....	26

Slika 6.3. Povezivanje PLC-a i HMI-a.....	26
Slika 6.4. Definirani PLC tagovi.....	28
Slika 6.5. Network 1.....	28
Slika 6.6. Network 4 i Network 5.....	29
Slika 6.7. Network 6 i Network 7.....	30
Slika 6.8. Network 8.....	30
Slika 6.9. Varijable u „Frekvencija=>Linearna_brzina_otv“ funkcijskom bloku.....	31
Slika 6.10. Funkcijski blok „Frekvencija=>Linearna_brzina_otv“.....	32
Slika 6.11. Network 9.....	34
Slika 6.12. Varijable u „Pozicioniranje_otvorena_petlja“ funkcijskom bloku.....	34
Slika 6.13. Funkcijski blok „Pozicioniranje_otvorena_petlja“.....	35
Slika 6.14. Network 14 i Network 15.....	38
Slika 6.15. Network 2.....	38
Slika 6.16. Network 10.....	39
Slika 6.17. Varijable u „Frekvencija=>Linearna_brzina_zat“ funkcijskom bloku.....	39
Slika 6.18. Funkcijski blok „Frekvencija=>Linearna_brzina_zat“.....	40
Slika 6.19. Network 12 i Network 13.....	41
Slika 6.20. Grafički prikaz odnosa v-t krivulje i prijednog puta.....	42
Slika 6.21. Rekonstrukcija signala ZOH metodom.....	42
Slika 6.22. Network 3, Network 11 i Network 16.....	43
Slika 6.23. Definirani HMI tagovi.....	44
Slika 6.24. Početni izborni zaslon.....	45
Slika 6.25. Zaslon za upravljanje u otvorenoj petlji.....	45
Slika 6.26. Zaslon za upravljanje u zatvorenoj petlji.....	46
Slika 6.27. Zaslon s trendom pozicije.....	46

Slika 6.28. Zaslone s trendom brzine.....	47
Slika 6.29. Početni izborni zaslon s označenim dijelovima.....	47
Slika 6.30. Zaslone za upravljanje u otvorenoj petlji s označenim dijelovima.....	48
Slika 6.31. Zaslone za upravljanje u zatvorenoj petlji s označenim dijelovima.....	49
Slika 6.32. Zaslone s trendovima pozicije i brzine.....	50
Slika 7.1. Odzivi brzina.....	51
Slika 7.2. Odzivi brzina u suprotnom smjeru.....	52
Slika 7.3. Odzivi pozicija.....	52

## Popis tablica

Tablica 3.1. Podaci asinkronog motora.....	7
Tablica 3.2. Podaci frekvencijskog pretvarača.....	14
Tablica 3.3. Funkcija korištenih priključaka u upravljačkom krugu.....	16
Tablica 3.4. Postavljeni parametri frekvencijskog pretvarača.....	17
Tablica 5.1. Osnovni simboli i operacije u ljestvičastom dijagramu.....	24
Tablica 6.1. Opis korištenih tipova podataka.....	27
Tablica 6.2. Vrijednosti linearnih brzina za odgovarajuću frekvenciju.....	33