

# Podešavanje i upravljanje Siemens Sinamics S120 pretvaračem preko web-sučelja

---

Dujmović, Fran

Master's thesis / Diplomski rad

2024

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:000736>

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

*Download date / Datum preuzimanja:* **2024-08-02**

*Repository / Repozitorij:*

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



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

**Sveučilišni studij**

**PODEŠAVANJE I UPRAVLJANJE SIEMENS SINAMICS  
S120 PRETVARAČEM PREKO WEB SUČELJA**

**Diplomski rad**

**Fran Dujmović**

**Osijek, 2024.**

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
1.1. Zadatak diplomskog rada .....	2
<b>2. PREGLED PODRUČJA TEME</b> .....	<b>3</b>
<b>3. SINKRONI STROJ</b> .....	<b>4</b>
<b>4. ANALIZA I KONFIGURACIJA PODSUSTAVA</b> .....	<b>6</b>
4.1. Frekvencijski pretvarač Siemens Sinamics S120 .....	6
4.2. Konfiguracija STARTER-a .....	12
4.3. Programabilni logički kontroler (PLC) .....	20
4.4. Konfiguriranje korištenjem TIA Portala.....	22
<b>5. IZRADA PROGRAMSKE LOGIKE UPRAVLJANJA MOTOROM</b> .....	<b>27</b>
5.1. PLC program za upravljanje motorom .....	28
5.2. HMI realizacija .....	38
<b>6. WEB SUČELJE</b> .....	<b>42</b>
6.1. Konfiguracija web sučelja .....	42
6.2. Korisnički definirane web stranice.....	47
<b>7. ZAKLJUČAK</b> .....	<b>57</b>
<b>LITERATURA</b> .....	<b>58</b>
<b>ŽIVOTOPIS</b> .....	<b>61</b>

# 1. UVOD

Industrijska automatizacija predstavlja jedan od najbitnijih čimbenika u modernom procesu proizvodnje. Gotovo svaka suvremena industrija zahtijeva automatizaciju procesa proizvodnje u svrhu povećanja vremenske učinkovitosti i efikasnosti samog proizvodnog procesa. Jedan od primjera gdje je pojam industrijske automatizacije neophodan jest prilikom upravljanja, parametriranja i puštanja u pogon asinkronih i sinkronih motora te servo-pogona. U tu svrhu razvijeni su frekvencijski pretvarači i programabilni logički kontroleri (PLC-ovi). Frekvencijski pretvarači su elektronički uređaji čija je primarna svrha transformacija izmjeničnog napona konstantne vrijednosti i frekvencije u izmjenični napon promjenjive vrijednosti i frekvencije sa ciljem upravljanja brzine vrtnje izmjeničnih motora. Osnovna podjela frekvencijskih pretvarača je na frekventne pretvarače sa i bez istosmjernog međukruga [1].

Programabilni logički kontroleri su industrijski računalni sustavi koji se koriste za automatizaciju procesa u industrijskim okruženjima. PLC-ovi služe za upravljanje i nadzor raznim industrijskim postrojenjima i strojevima, a sastoje se od procesorske jedinice, ulazno-izlaznih i komunikacijskih modula. Prednosti korištenja PLC-a uključuju brzu i fleksibilnu rekonfiguraciju sustava, pouzdanost, otpornost na elektromagnetske smetnje, mogućnost nadzora i dijagnostike te mogućnost integracije sa drugim industrijskim sustavima. PLC-ovi su postali standard u industrijskoj automatizaciji i omogućuju optimizaciju procesa, poboljšanje učinkovitosti i smanjenje troškova proizvodnje.

U diplomskom radu koristi se Siemens Sinamics S120 frekvencijski pretvarač sa ugrađenim web serverom u svrhu izrade web sučelja za daljinsko upravljanje izmjeničnim motorom. Siemens S120 koristi se za upravljanje i kontrolu brzine elektromotora u industrijskim aplikacijama. Izmjenični motor je spojen na PLC pomoću čega je izrađena logika upravljanja motora koristeći softversko okruženje TIA Portal dok je korisničko sučelje izrađeno koristeći HTML, CSS i JavaScript.

## **1.1. Zadatak diplomskog rada**

Glavni zadatak diplomskog rada jest dizajniranje, konfiguracija i implementiranje interaktivnog korisničkog sučelja za analitički i grafički prikaz parametara i performanse izmjeničnog motora. Web sučelje također mora služiti za upravljanje motorom te omogućiti potrebne sigurnosne mjere prilikom korištenja motora. Logiku upravljanja samog motora je potrebno implementirati na način da se omogući pokretanje i zaustavljanje motora te vrtnja motora pri raznim brzinama. Također, upravljanje motorom je moguće kroz razrađeno HMI sučelje koristeći PLC.

## 2. PREGLED PODRUČJA TEME

Diplomski rad temelji se na izradi korisnički definiranog web sučelja za upravljanje i prikaz parametara izmjeničnog motora čiji je princip funkcioniranja i logika upravljanja razrađena u softverskom okruženju TIA Portal.

U [1, 2] definiran je pojam i podjela frekvencijskih pretvarača, zajedno sa načinom funkcioniranja i primjenom istih u procesu industrijske automatizacije. Frekvencijski pretvarač korišten u diplomskom radu spada u kategoriju frekvencijskih pretvarača sa izmjeničnim međukrugom.

Literatura [3] opisuje konstrukciju i princip djelovanja asinkronih i sinkronih strojeva u ovisnosti o napajanju, frekvenciji te teretu stroja.

Na službenoj Siemens web stranici [4] dana su najnovija poboljšanja u radu firmware i hardware frekvencijskog pretvarača Siemens Sinamics S120 koji se koristi u diplomskom radu, sa pojašnjenjima integracije web poslužitelja u cilju poboljšanja funkcije dijagnostike i održavanja ili servisiranja pretvarača.

Korisnički priručnik [5] pruža sve informacije, postupke i radnje operatera potrebne za određenu fazu korištenja te daje princip spajanja upravljačke jedinice pretvarača sa kontrolerom.

Na web stranici [6] dat je alatni okvir za simulaciju i upravljanje raznim elektromotorima u ovisnosti o pretvaračima te prikazuje shematski prikaz osnovnog linijskog modula pretvarača. Web stranica [7] predstavlja bazu znanja, odnosno set potrebnih dokumenata o Siemens proizvodima te je dat blok dijagram načina povezivanja linijskog i motornog modula pretvarača.

Korisnički priručnik [8] za konfiguraciju pogona daje podršku vezanu za odabir motora, proračun pogonskih komponenti, odabir potrebnih dodataka kao i pri odabiru mrežnih i motornih opcija napajanja, kao što je prikaz spajanje motora sa enkoderom i pretvaračem putem DRIVE-CliQ sučelja. Literatura [9] objašnjava principe rada i sastav PLC uređaja.

Web stranica [10] daje tehničke podatke PLC uređaja koja se koristi u diplomskom radu.

Literatura [11] objašnjava način povezivanja PLC uređaja sa softverskim okruženjem TIA Portal.

Korisnički priručnik [12] objašnjava konfiguraciju i implementiranje programskog koda za kreiranje korisnički definiranih web stranica.

### 3. SINKRONI STROJ

Glavno obilježje električnih strojeva jest mogućnost elektromehaničke pretvorbe energije. S obzirom na smjer pretvorbe energije osnovna podjela električnih strojeva je na generatore i motore, odnosno generatori pretvaraju mehaničku energiju u električnu dok motori električnu energiju pretvaraju u mehaničku. Dovođenje električne energije stroju rezultira motorskim režimom rada stroja, dok suprotno djelovanje rezultira generatorskim režimom rada stroja. Osnovni fizikalni izrazi koji opisuju ponašanje električnih strojeva su sljedeći:

1. Faradayev zakon elektromagnetske indukcije (električni napon kao funkcija promjene magnetskog toka):

$$\Delta \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (3-1)$$

2. Ampereov zakon protjecanja (magnetski napon kao funkcija gibanja naboja)

$$\Delta \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t} \quad (3-2)$$

3. Lorentzov zakon sile na vodič u magnetskom polju (gibanje naboja kroz vodič u magnetskom polju)

$$\vec{F}_{mag} = I (\vec{l} \times \vec{B}) \quad (3-3)$$

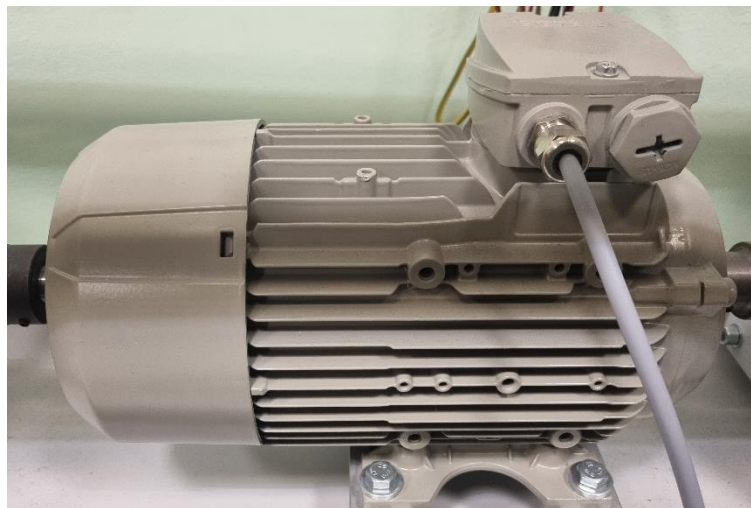
Osnovna konstrukcija električnih strojeva sastoji se od statora (stacionarnog dijela) i rotora (pomičnog dijela). Stator i rotor imaju zasebne namote koji služe za stvaranje magnetskog toka (uzbude) i induciranje napona (armature – namot na kojem se stvara sila zakreta). Pri izmjeničnom izvoru napajanja najčešće se koriste asinkroni i sinkroni strojevi. Ključna razlika između navedenih strojeva jest ta da sinkroni stroj rotira sinkronom brzinom, odnosno brzina vrtnje rotora jednaka je brzini vrtnje okretnog magnetskog polja induciranog statorskim strujama [3]:

$$n = n_s = \frac{60 \cdot f}{p} \quad (3-4)$$

- $n_s$  – sinkrona brzina [ $min^{-1}$ ]
- $f$  – frekvencija napona napajanja [Hz]
- $p$  – broj pari polova

Namot na statoru sinkronog stroja naziva se armaturni namot. Struja koja prolazi armaturnim namotom u magnetskom polju uzbuđnog namota uzrokuje pojavu magnetske sile na vodiče armaturnog namota. Analogno tome namot na rotoru sinkronog stroja naziva se uzbuđni namot kroz koji prolazi nepromjenjiva struja koja uzrokuje stvaranje uzbuđnog magnetskog toka na rotoru sinkronog stroja. Posebna izvedba sinkronih strojeva su sinkroni strojevi sa permanentnim magnetima. Razlika u te dvije izvedbe je u tome da se sa permanentnim magnetima dobiva nepromjenjiva uzbuda dok se kod izvedbe sa namotima može mijenjati iznos uzbude, što je bitno kod većih snaga i promjena tereta. Bitno je naglasiti kako kod sinkronih strojeva ne postoji razlika između brzine vrtnje okretnog magnetskog polja statora i brzine vrtnje rotora (ne postoji klizanje), ali postoji problem kako postići da se pokrene mehanička masa rotora i dostigne okretno magnetsko polje (pokretanje sinkronih motora) te problem da ne dođe do zaostajanja rotora (ispad iz sinkronizma).

U diplomskom radu koristi se sinkroni motor Siemensove serije 1FK7. Motori navedene serije spadaju u grupu sinkronih motora sa permanentnim magnetima te imaju (u svrhu upravljanja) ugrađen apsolutni enkoder za precizno praćenje pozicije i brzine vrtnje motora uz digitalnu pretvorbu izlaznih signala prilagođenu za pretvarač, odnosno nije potrebna kalibracija referentne točke. U cilju povećanja sigurnosti rada motora pod opterećenjem u statorskom namotu je ugrađen KTY 84 temperaturni senzor za mjerenje promjene temperature motora.



Slika 3.1 Siemens 1FK7 sinkroni motor s permanentnim magnetima

Zbog održavanja stabilnosti motora te same konstrukcije pogona, brzina vrtnje neće prekoračiti  $100 \text{ min}^{-1}$ . U daljnjoj razradi biti će detaljno objašnjen princip spajanja, konfiguracije i puštanja u pogon navedenog motora.



## 4. ANALIZA I KONFIGURACIJA PODSUSTAVA

Prije realizacije zadataka diplomskog rada potrebno je zasebno teorijski razraditi princip funkcioniranja i implementaciju svakog pojedinog elementa pogona za analitički i grafički prikaz parametara i performanse izmjeničnog motora kojim se želi postići upravljanje kombinacije sinkronog motora i pretvarača. Pri tome treba razlikovati softverske i hardverske konfiguracije koje zajedno omogućuju upravljanje motorom.

### 4.1. Frekvencijski pretvarač Siemens Sinamics S120

Siemens S120 je pretvarač koji se koristi za kontrolu brzine i pozicije elektromotora u industrijskim aplikacijama te omogućava modularni sustav pretvarača i motora, odnosno dizajn Sinamics S120 osigurava visoku preciznost, pouzdanost i fleksibilnost u upravljanju različitim vrstama motora zbog univerzalnosti osnovnih hardware komponenti i prilagodljivosti programske podrške.

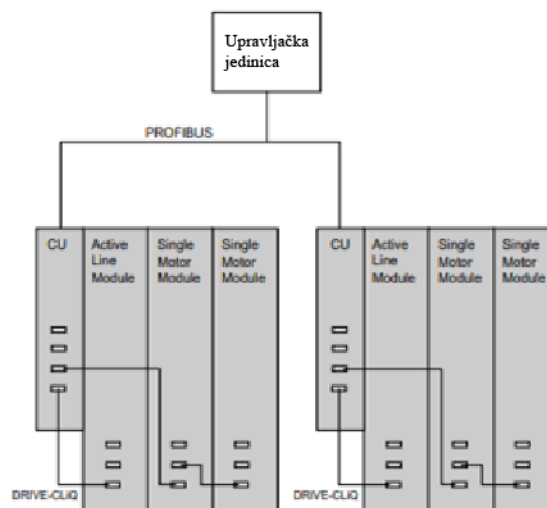


Slika 4.1 Modularni frekvencijski pretvarač Siemens S120 [4]

Komponente Sinamics S120 pogonskog sustava podijeljene su u tri kategorije;

- Kontrolna jedinica (engl. "*Control Unit*")
- Linijski modul (engl. "*Line Module*")
- Jedan ili više serijski spojenih invertera modula motora (engl. "*Motor Module*")

Kontrolna jedinica (CU) je glavna komponenta pogonskog sustava zaslužena za upravljanje i nadzor pogonskog sustava. Ona omogućuje korisniku programiranje i konfiguraciju parametara pogonskog sustava, uključujući podešavanje brzine, momenta, kontrolnih petlji, sigurnosnih postavki i drugih parametara koji su specifični za aplikaciju. Kontrolna jedinica ima ugrađeno sučelje preko kojeg korisnik može upravljati i nadzirati pogonski sustav. Preko sučelja se prikazuju relevantni podaci poput brzine motora, stanja rada, alarmnih poruka i drugih dijagnostičkih podataka. Bitno je naglasiti kako kontrolna jedinica podržava različite protokole komunikacije kako bi se omogućila integracija s drugim sustavima. To može uključivati protokole poput PROFINET, PROFIBUS, EtherCAT i drugih protokola. Komunikacija omogućuje razmjenu podataka između pogonskog sustava (pretvarač i motori) i drugih uređaja nadređene razine automatizacije, kao što su PLC ili SCADA (engl. „*Supervisory Control and Data Acquisition*“) sustavi, koji mogu nadzirati rad više pogonskih sustava (npr. industrijski procesi sa više linijski povezanih faza koji se zasebno upravljaju u cilju istovremenosti rada procesa).



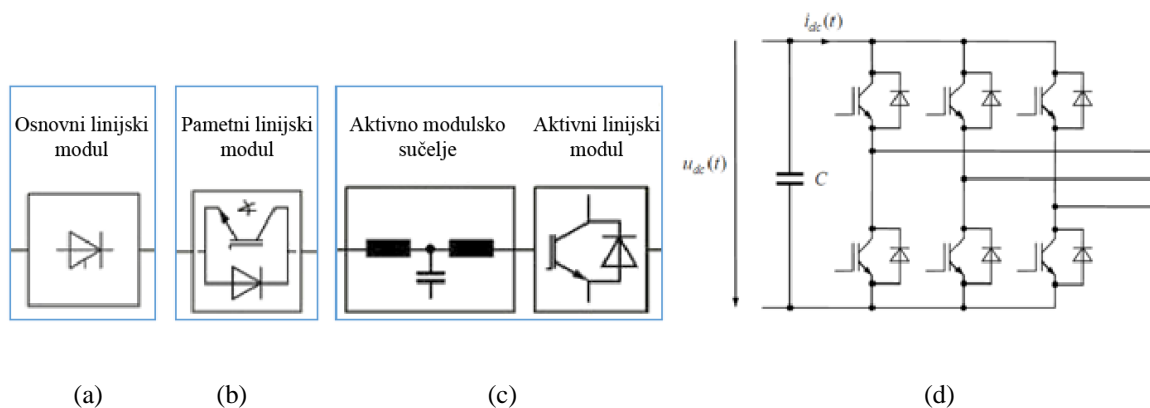
Slika 4.2 Shema spajanja kontrolne jedinice sa kontrolerom koristeći PROFIBUS standard [5]

Na slici 4.2 prikazana je shema konfiguracije kontrolne jedinice pogonskog sustava Siemens S120 sa kontrolerom i više spojenih jedinica modula pretvarač/motori. Kontroler (npr. PLC) i kontrolne jedinice dva pretvarača u prikazanom primjeru povezani su koristeći PROFIBUS standard, putem kojeg se prenose informacijski bitovi u oba smjera komunikacije (stanje pogona ↔ analiza i upravljački signal) od kontrolera do kontrolnih jedinica. Aktivni linijski modul i kontrolna jedinica povezani su sa sučeljem DRIVE-CLIQ koje omogućuje komunikaciju i razmjenu podataka unutar istog pogonskog sustava. U posebnim slučajevima se pogonski sustavi mogu povezati i bez kontrolera (jednostavnije upravljanje serijskih pogona).

Linijski modul služi za ispravljanje ulaznog trofaznog napajanja u zajedničku istosmjernu (DC) sabirnicu te na taj način napaja motorne module. Prilikom priključenja napajanja struja proteče kroz istosmjerni međukrug. Linijski moduli se također nazivaju "infeed" moduli te se dijele na sljedeće kategorije:

- Osnovni (engl. „Basic“) infeed modul
- Smart infeed modul
- Aktivni infeed modul

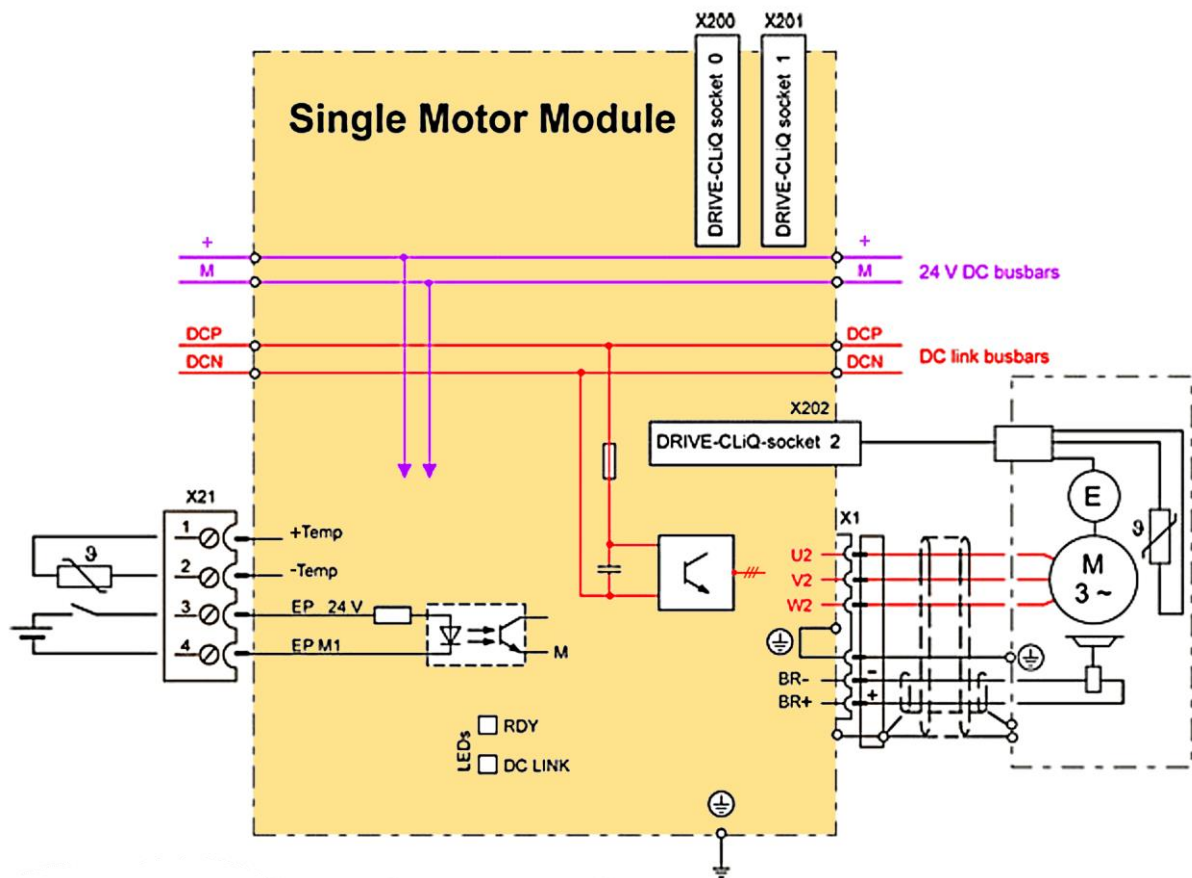
Osnovni linijski modul predstavlja naponski ispravljač koji ispravlja ulazni trofazni napon u istosmjerni napon te pritom napaja istosmjerni međukrug i motorne module. Smjer toka energije pri korištenju ovog tipa modula jest od pojne mreže prema trošilu. U slučaju generatorskog režima rada gdje je smjer toka energije obrnut koristi se poseban modul kočionog otpornika za potrebnu disipaciju energije u istosmjernom međukrugu. Smart i Aktivni moduli osiguravaju mogućnost povrata energije iz motora u mrežu (generatorsko kočenje motora) uz dodatnu mogućnost aktivnih modula da reguliraju i razinu DC međukruga u svrhu bolje efikasnosti rada pogona.



Slika 4.3 Shematski prikazi: (a) osnovnog, (b) smart i (c) aktivnog linijskog modula te (d) motornog modula [6]

Na slici 4.3 prikazani su blok prikazi tehnologija rada različitih verzija ispravljača (osnovni = jednosmjerni, smart i aktivni = dvosmjerni) pri čemu tehnologija aktivnog modula omogućava i upravljanje razinom DC napona. Zbog redukcije valovitosti izlaznog DC napon uobičajeno se koristi 12 pulsna topologija čime se smanjuje opterećenja filtera DC kruga. Analogno tome na izlazu prema motorima koriste se izmjenjivači koji su u pravilu dvosmjerni, odnosno osiguravaju povrat energije iz motora u DC međukrug, gdje se onda ili dodaju otpori za trošenje povratne energije (upravljanje kočenjem motora) ili filteri prema dvosmjernim ispravljačkim ulaznim modulima (smart ili aktivni) za povećanje efikasnosti (generatorski rad motora).

Motorski moduli upravljaju motorom pretvarajući ulazno istosmjerno napajanje iz istosmjernog međukruga u trofazno izmjenično napajanje koristeći pulsno-širinsku modulaciju.



Slika 4.4 Blok dijagram povezivanja motornog modula [7]

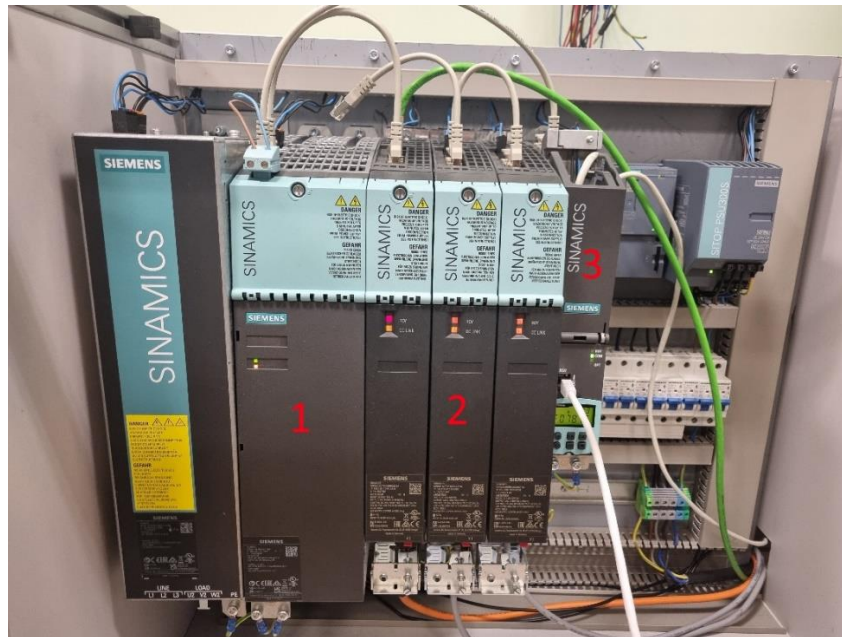
Kako je prethodno naglašeno linijski (*infeed*) moduli napajaju motorne module koristeći struju istosmjernog međukruga (engl. „*DC Link*“). Jedinični modul motora služi za spajanje jednog motora na modul dok se dvostruki modul koristi za spajanje više motora na isti modul (osigurava se moguća razmjena energija između motora preko DC linka, gdje iz mreže dolazi samo potrebna razlika energije).

U diplomskom radu konekcija između frekventijskog pretvarača sa ostalim podsustavima ostvarena je koristeći PROFINET standard. Na upravljačkoj jedinici nalaze se tri priključka (engl. "*Port*"):

1. X127 priključak
2. X150 P1 priključak
3. X 150 P2 priključak

Svaki priključak koji se koristi mora imati dodijeljenu IP adresu putem PG/PC sučelja. Putem dodijeljenih IP adresa omogućuje se konekcija između priključaka i upravljačkog sustava.

Glavni priključak za puštanje u pogon frekvencijskog pretvarača je X127, no u diplomskom radu koristio se X150 P1 priključak za realizaciju sigurnosnih protokola.

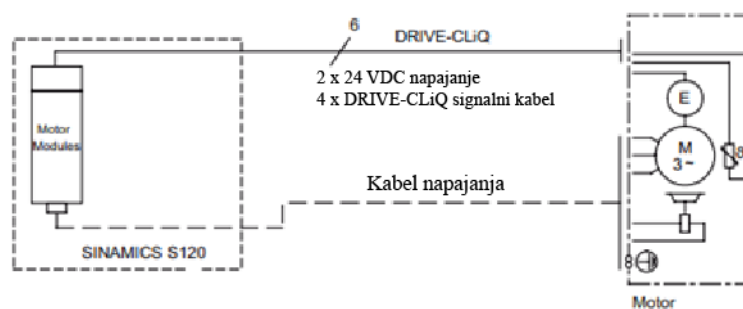


Slika 4.5 Siemens S120 frekvencijski pretvarač

Kompletan frekvencijski pretvarač korišten u diplomskom radu prikazan je na slici 4.5 te su na istoj označeni sljedeći dijelovi:

1. Aktivni linijski modul
2. Motorni modul
3. Upravljačka jedinica

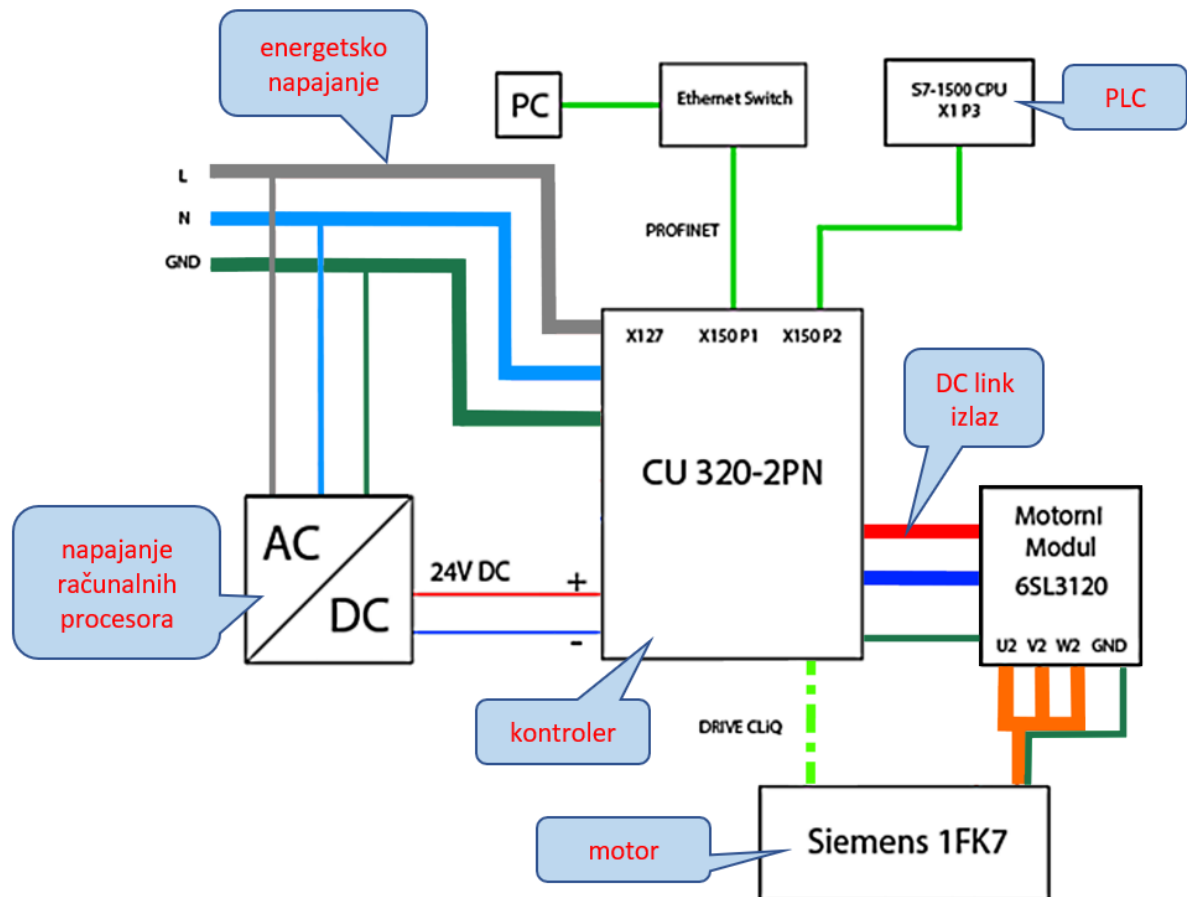
Prethodno spomenuto DRIVE-CLiQ sučelje (prikazano na slici 4.5. zelenim kablom) spaja motorni modul sa sinkronim motorom.



Slika 4.6 Shematski prikaz spoja motora sa frekvencijskim pretvaračem [8]

Konfiguracija sklopovlja prikazana na slici 4.6 omogućuje direktan prijenos digitalnih signala te obrađivanja istih pomoću apsolutnog enkodera.

Slika 4.7 prikazuje ukupnu shemu međusobnog spajanja svog sklopovlja za laboratorijski sklop koji se obrađuje u ovom radu. Upravljači signalni vodovi spajaju računalo (PC) na Ethernet sklopku putem PROFINET veze (Ethernet standard), dok je sinkroni motor spojen na upravljačku jedinicu putem DRIVE-CliQ sučelja.

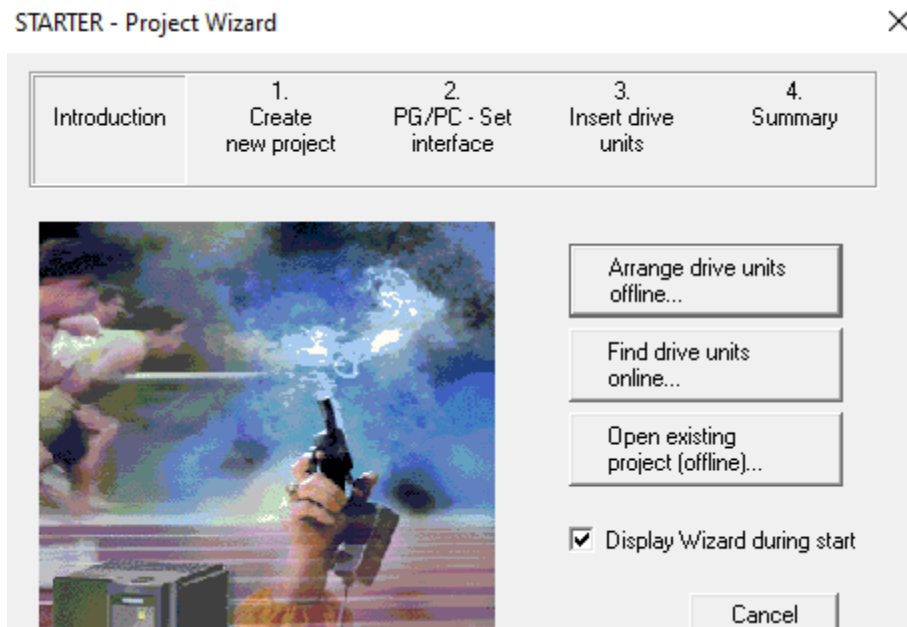


Slika 4.7 Ukupna shema spajanja sklopovlja

PROFINET vezom je spojen izlaz X150 P1 upravljačke jedinice na spomenutu Ethernet sklopku, dok priključak X150 P2 služi za povezivanje X1 P3 priključka programibilnog logičkog kontrolera (PLC) sa upravljačkom jedinicom. Bitno je naglasiti kako mora biti korišten jedan od Ethernet standarda za uspostavljanje komunikacije sa PLC-om i frekvencijskim pretvaračem (u ovom slučaju je to PROFINET). Svi podaci, firmware, softverske promjene, licence i slično je pohranjeno u memorijskoj kartici upravljačke jedinice. Energetski vodovi spajaju AC mrežu fiksne frekvencije sa kontrolerom (ispravljač + filter + DC link), izlaz prema motornom modulu u obliku DC vodova te u konačnici energetski izlaz iz motornog modula (izmjenjivač) ide u motor u obliku AC vodova kojim teče kontrolirano AC energetsko napajanje. U daljnjoj razradi biti će objašnjen postupak pripreme softvera za puštanja u pogon te princip djelovanja i konfiguracije korištenog PLC-a.

## 4.2. Konfiguracija STARTER-a

Prije pokretanja samog motora potrebno je prvo obaviti puštanje u pogon samog stroja na način da se konfiguriraju svi potrebni parametri i sučelje motora (engl. "*Commissioning*"). U tu svrhu razvijen je softverski alat od strane Siemens-a zvan STARTER.



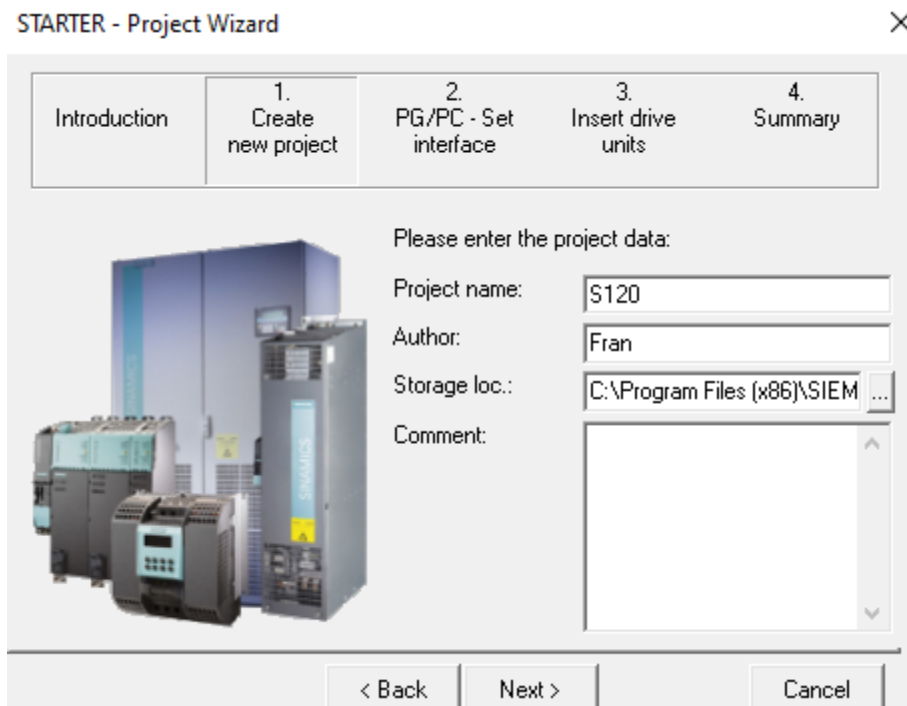
Slika 4.8 Početno sučelje STARTER-a za odabir načina implementacije frekvencijskog pretvarača

Za početak korištenja STARTER-a potrebno je dodati programske module - komponente frekvencijskog pretvarača unutar samog softvera. Postoje dva načina za implementaciju frekvencijskog pretvarača u STARTER program;

1. Ručno dodavanje zasebnih komponenata i parametara pretvarača (*'Offline'*)
2. Automatsko dodjeljivanje komponenata i parametara pretvarača kroz LAN vezu između softvera i pretvarača (*'Online'*)

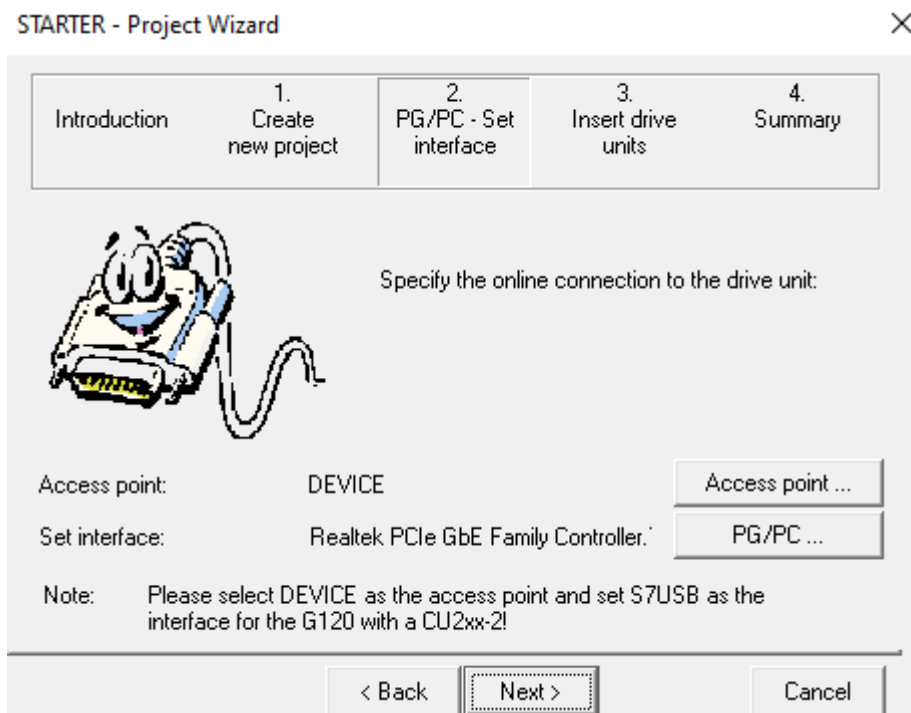
Na slici 4.7 prikazano je kako je upravljačka jedinica frekvencijskog pretvarača PROFINET vezom povezana sa računalom kroz Ethernet sklopku (najčešće kod izgradnje složenijih upravljačkih mreža), a moguće je i direktno povezati frekvencijski pretvarač i računalu kroz dostupne priključke. Prilikom pokretanja samog frekvencijskog pretvarača te odabir opcije automatskog pronalaženja frekvencijskog pretvarača (engl. "*Drive*") softver pronalazi i dodjeljuje potrebne komponente unutar softvera.

Nakon što je kreirana mapa sa svim potrebnim komponentama kreće se u kreiranje projekta te unos naziva projekta.



Slika 4.9 Kreiranje projekta unutar STARTER programa

Omogućeno je mijenjanje lokacije na kojoj će datoteka softvera biti spremljena. Ovaj korak će biti potreban kasnije za implementaciju softverske konfiguracije sa PLC-om.

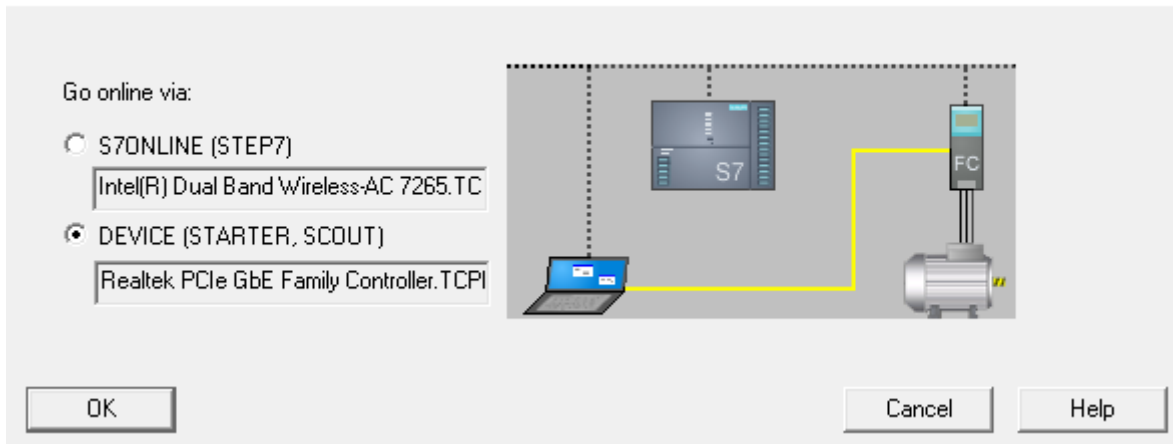


Slika 4.10 Postavljanje PG/PC sučelja



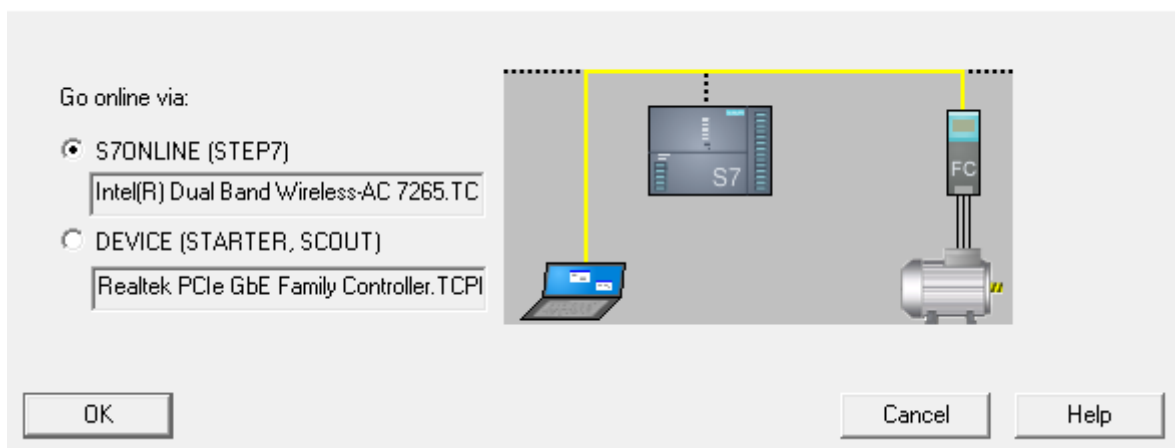
Konfiguriranje PG/PC sučelja prikazano je na slici 4.10 gdje se definira komunikacija između PLC uređaja i računala putem predefiniраниh mrežnih čvorova. Spajanje navedenih sustava moguće je na dva načina: direktno i preko STEP7 sučelja.

#### Set Access Point for Accessible Nodes



Slika 4.11 a) Direktno povezivanje računala sa frekvencijskim pretvaračem

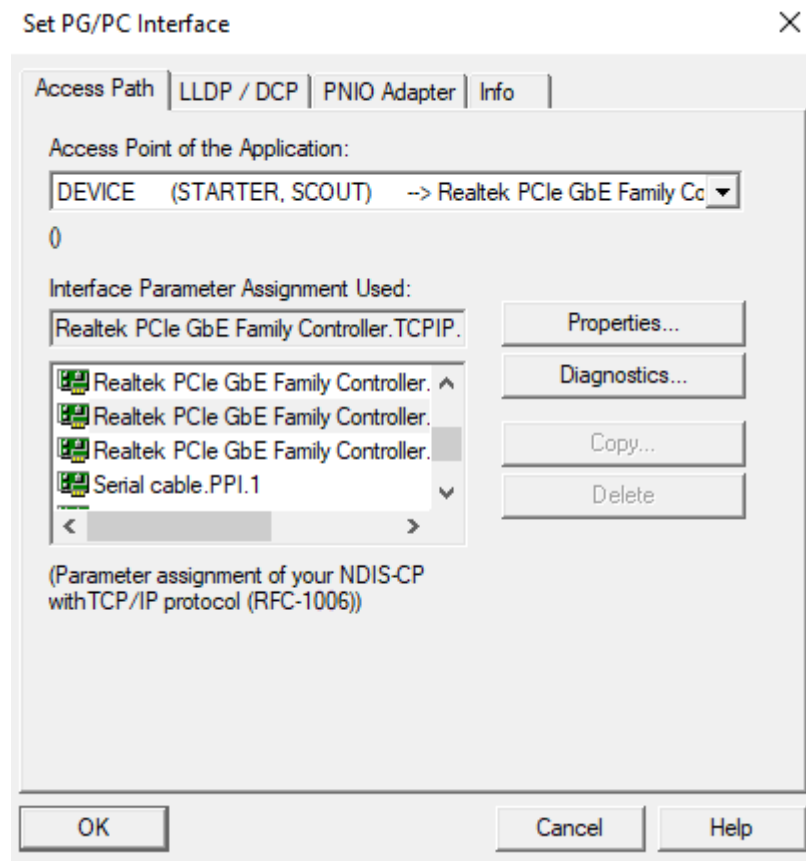
#### Set Access Point for Accessible Nodes



Slika 4.11. b) Povezivanje računala preko STEP7 sučelja sa frekvencijskim pretvaračem

Odabran je direktni način povezivanja računala sa frekvencijskim pretvaračem. PG/PC sučelje odabire se na način da se izabere parametar sučelja koji odgovara stvarnom spoju. Komunikacija se zatim omogućava kroz uspostavljeni TCP/IP protokol, kako je prikazano na slici 4.12.

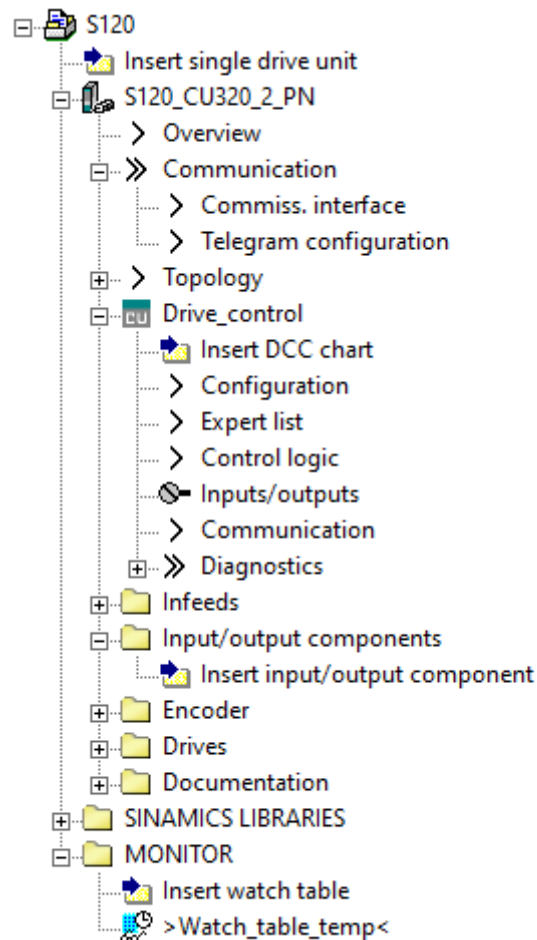
Softverski program STARTER automatski pronalazi sve dostupne komponente frekvencijskog pretvarača koje su spojene na računalo. Kako je prethodno rečeno, u diplomskom radu priključak koji služi za povezivanje pretvarača sa računalom kroz Ethernet sklopku je X150 P1. IP adresa priključka pretvarača postavljena je na predefiniranu vrijednost kroz PG/PC sučelje te iznosi 192.168.0.2. Ovaj korak je iznimno bitan za ispravnu uspostavu komunikacije između pretvarača i PLC-a te za konfiguraciju web servera.



Slika 4.12 Odabir parametara sučelja

Svaki priključak pretvarača ima zasebno predefiniranu IP adresu koju je moguće kasnije promijeniti ukoliko se potreba za time pojavi. Također je potrebno postaviti IPv4 adresu PLC uređaja kroz softverski alat TIA Portal.

U softverskom sučelju STARTER nakon definiranja svih prethodnih koraka može se prikazati mapiranje svih datoteka i mapa koje zajedno čine parametarski model potreban da se može provoditi signalno povezivanje fizičkih modula.



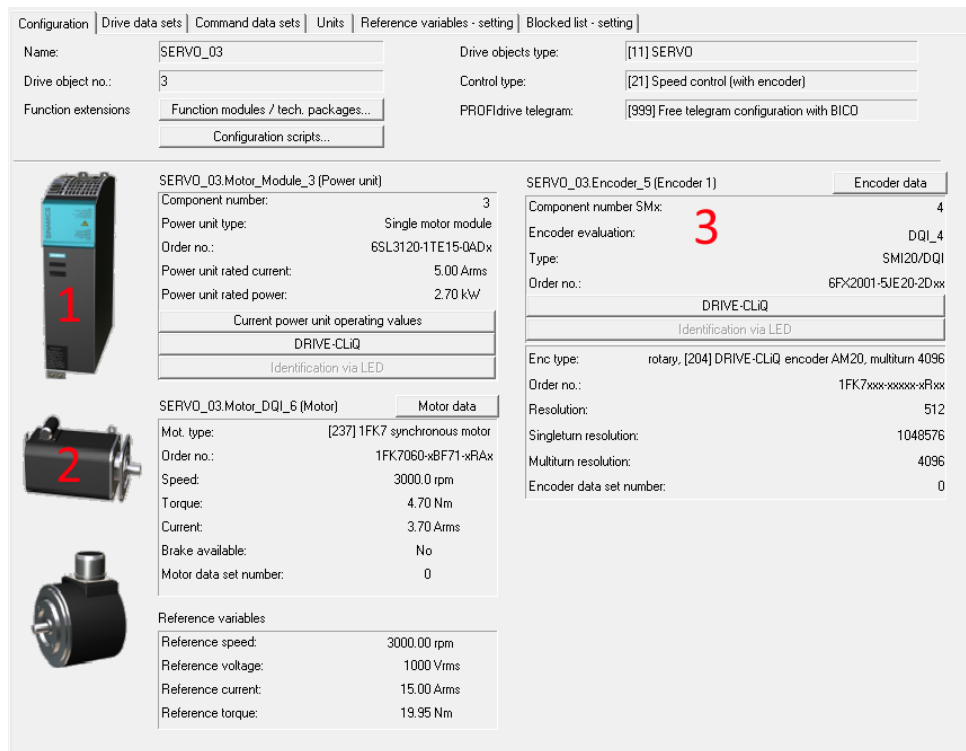
Slika 4.13 Siemens Sinamics S120 frekvencijski pretvarač u softveru STARTER

Nakon uspješnog postavljanja PG/PC sučelja i odabira pristupnih mrežnih čvorova STARTER automatski pronalazi frekvencijski pretvarač te dodjeljuje sve komponente unutar softvera. Parametriranje komponenata pretvarača, dodavanje motora, enkodera, infeed modula i sličnog vrši se odabirom;

1. *"Download to selected target device"*
2. *"Automatic configuration"*

Odabirom automatske konfiguracije STARTER prepoznaje predefinirane pogonske sustave i dodjeljuje sve komponente koje su povezane na frekvencijski pretvarač te postavlja predefinirane parametre.

Na prikazu STARTER sučelja nakon provedenih koraka prikazuju se vizualne komponente koje je program odabrao sa njihovim osnovnim parametrima kako je prikazano na slici 4.14.

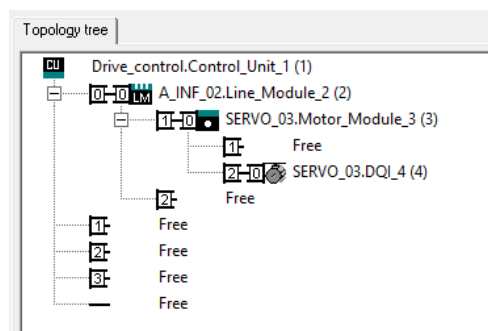


Slika 4.14 Grafički prikaz spojenih komponenti na frekvencijski pretvarač unutar STARTER-a

Koristeći opciju automatske konfiguracije parametara pretvarača pronađeni su i dodani sljedeći parametri;

1. Motorni modul 6SL3120
2. Sinkroni motor 1FK7
3. Enkoder 6FX2001

Parametri svih navedenih komponenti spojeni na pretvarač automatski su dodijeljeni korištenjem opcije "Configure with Wizard".



Slika 4.15 Topologija dodanih komponenata

Uspostavljena komunikacija između softvera i pretvarača prikazane na slici 4.16 odvija se kroz prethodno spomenuti X150 P1 priključak putem IP adrese. Kroz istu IP adresu biti će uspostavljen i konfiguriran web server S120 frekvencijskog pretvarača.

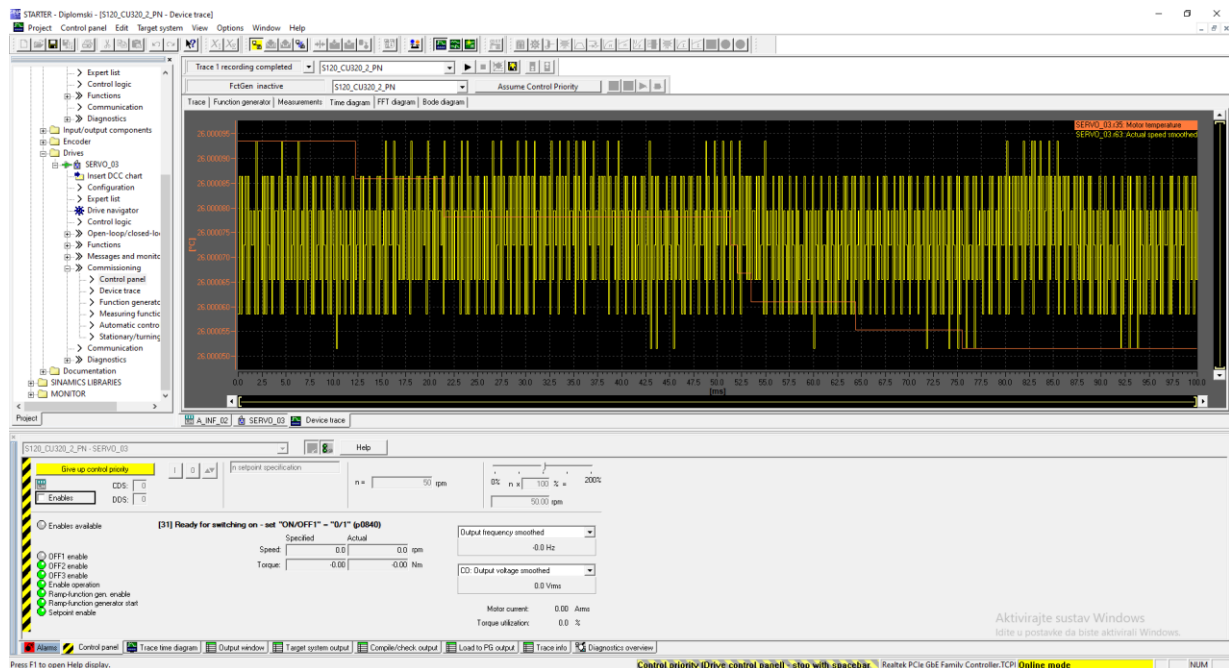


Slika 4.16 Sučelje puštanja u pogon

Nakon što su ispravno detektirane i dodijeljene komponente frekvencijskog pretvarača i ostalih sustava unutar STARTER-a te je provedena adekvatna parametrizacija upravljačkog sustava moguće je koristiti "*Device Trace*" funkciju za grafički prikaz odabranih signala motora. Potrebno je odabrati opciju "*Assume Control Priority*" unutar "*Control Panel*" podsustava upravljačkog panela. STARTER preuzima kontrolu nad motornim sustavom te omogućuje pokretanje samog motora. Potrebno je omogućiti opciju "*Enables*" i postaviti određenu brzinu vrtnje motora, u ovom slučaju brzina motora postavljena je na  $n = 50 \text{ rpm}$ .

Ovim korakom završena je konfiguracija puštanja u pogon sinkronog motora.

Motor se počinje okretati zadanom brzinom i grafički se prikazuju odabrani signali (slika 4.17). Na zaslonu STARTER-a programski se prilagođava prikaz promjenom veličina. U donjem dijelu prikaza na slici 4.17 mogu se uočiti određeni sigurnosti pokazivači.



Slika 4.17 Pokretanje motora i grafički prikaz signala temperature i brzine vrtnje motora

Zadanu konfiguraciju potrebno je spremati te istu prenijeti na PLC radi ispravne uspostave komunikacije između PLC-a i Drive-a. Opcija "*Enables*" aktivira linijske module pretvarača.

U prvom pokušaju programiranja i puštanja u rad pojavio se problem u obliku da je program detektirao signal za aktivaciju uključenja linijskog modula kao aktivan, odnosno zadan prije aktivacije samog linijskog modula, stoga nije bilo moguće ručno preuzeti kontrolu nad sustavom.

Razlog tomu je taj što je na Drive-u bila detektirana prijašnja konfiguracija Drive-a koja je bila spremljena u ROM memoriju te su vrijednosti bitova bile u međusobnom konfliktu, odnosno parametar je bio trajno uključen. Parametar koji je odgovoran za memorijsku alokaciju i upravljanje bitovima za aktivaciju, odnosno deaktivaciju linijskog modula jest parametar oznake "*r46*".

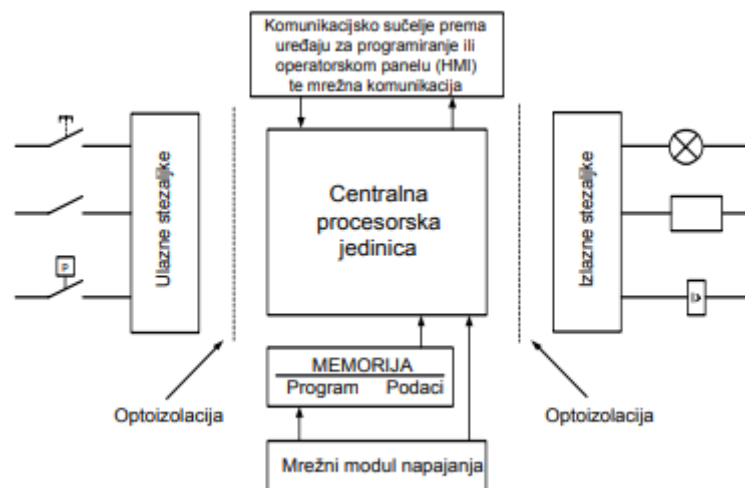
Rješenje je bilo resetirati vrijednost parametra "*p10*" koji služi kao filter za puštanje u pogon linijskog modula na vrijednost "*p10 = 0*" te ponovna konfiguracija podsustava. Time su se oba parametra resetirala na početne vrijednosti i omogućeno je preuzimanje kontrole nad sustavom.

### 4.3. Programabilni logički kontroler (PLC)

U svrhu unapređenja automatizacije industrijskih proizvodnih procesa kao zamjena za mreže elektromehaničkih releja (mnoštvo fizičkih elemenata i spojnih žica) krajem šezdesetih godina prošlog stoljeća uvode se elektronički uređaji koji koristeći mogućnosti industrijskog računala (male dimenzije, memorija, procesor) i sklopova energetske elektronike (industrijski sklopnici, ulazi, izlazi) objedinjeni u jedan programski upravljivi sklop kojim se omogućuju softversko upravljanje digitalnim (signalnim) veličinama koje ona osiguravaju upravljanje analognim (energetskim) tokovima, popularno nazvan programabilni logički kontroler (PLC).

Osnovna konfiguracija PLC-a se sastoji od sljedećih komponenti:

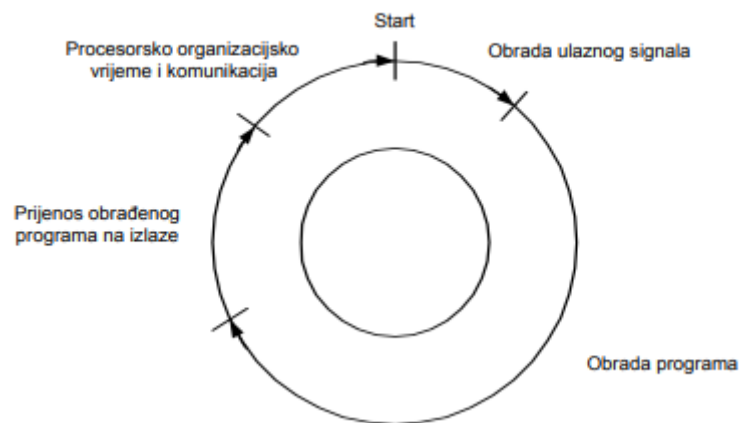
- Ulazno/Izlazne jedinice (I/O)
- CPU jedinica (procesorska jedinica)
- Modul napajanja
- Moduli za proširenje



Slika 4.18 Dijelovi PLC-a [9]

Temeljni dio svakog PLC-a jest centralna procesorska jedinica (CPU). Ona je zaslužena za obradu, procesiranje i upravljanjem svih informacijskih signala koji se prenose kroz PLC. Korištenjem elektroničkih sklopova CPU jedinica "čita stanja" analognih i digitalnih ulaza, izvršava isprogramiranu logiku putem koda te prosljeđuje određene izlazne digitalni signale koji se opet putem određenih elektroničkih sklopova prosljeđuju na izlaze PLC-a. Kompletni PLC i svi ostali moduli napajaju se putem standardnog modula napajanja 24V DC. Osnovni princip funkcioniranja PLC-a je ciklička obrada podataka u beskonačnoj petlji koja se ponavlja za svaku inicijalizaciju programa [9], ali je moguće i zadavati vremenski ili brojčani broj ciklusa.

Princip rada programabilnog logičkog kontrolera pojašnjen je na slici 4.19. PLC vrši program ciklički u beskonačnoj petlji koja se sastoji od gore navedenih faza. U prvoj fazi vrši se obrada ulaznog signala gdje PLC očitava ulazne vrijednosti i početna stanja zadanih varijabli te sprema iste u zasebno alociranu memoriju unutar procesorske jedinice. Prije početka druge faze cikličke obrade mora biti jasno definirana logika upravljanja procesa.



Slika 4.19 Ciklus rada PLC-a [9]

Logiku upravljanja procesom moguće je realizirati na tri zasebna načina – "*Ladder*" dijagramom (LAD), funkcijskim blokovskim dijagramom (FBD) te strukturnim kontrolnim jezikom (SCL). U drugoj fazi cikličke obrade signala PLC izvršava programsku logiku te rezultate pohranjuje u predefiniranim memorijskim adresama procesorske jedinice. Trećom fazom rada PLC-a smatra se proces prijenosa obrađenog programa na fizičke izlaze PLC-a. Četvrta i zadnja faza odnosi se na komunikaciju PLC-a sa vanjskim jedinicama u određenom periodu vremena.

Ulazi programa su uvijek označeni sa "*I*" dok su izlazi označeni sa "*Q*". Tijekom izvođenja programske logike česta pojava zna biti da izlaz u isto vrijeme poprimi više signala za aktivaciju, rezultirajući trajnim uključenjem zadanog izlaza. Kako bi se zaobišao taj problem uvode se pomoćne memorijske varijable označene kao "*%MX.Y*" u memorijskom registru. Ovakav način implementacije memorijskih varijabli omogućuje postavljanje memorijskih varijabli na izlaze umjesto varijabla za fizički izlaz koje se mogu detaljnije mijenjati i manipulirati.



U diplomskom radu koristi se procesorska jedinica prikazana na slici 4.20. Priključak na slici služi za povezivanje frekvencijskog pretvarača sa PLC-om koristeći PROFINET. Nakon ispravnog povezivanja PLC-a sa računalom potrebno je softverski konfigurirati uređaj koristeći Siemensovo razvojno okruženje TIA Portal.



Slika 4.20 CPU 1510SP F-1 PN [10]

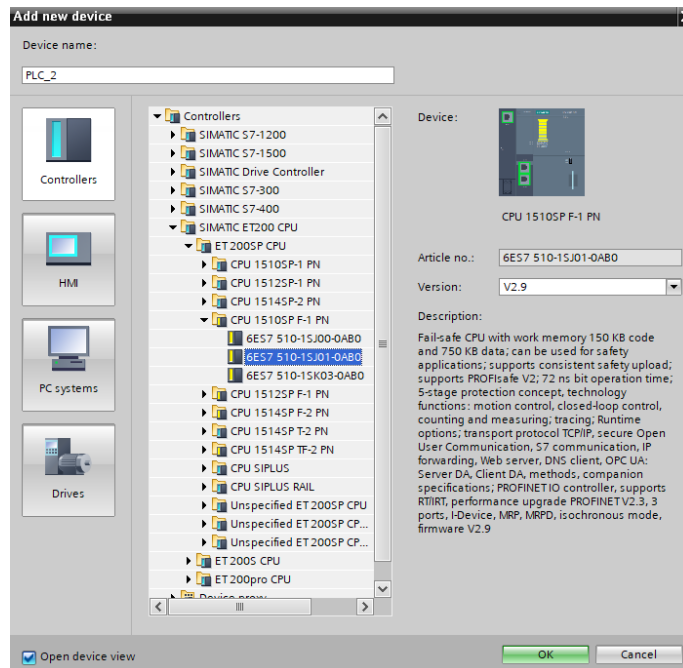
#### **4.4. Konfiguriranje korištenjem TIA Portala**

TIA Portal (engl. „*Totally Integrated Automation Portal*“) je programsko razvojno sučelje razvijeno od strane Siemens-a koje se koristi za izradu i implementaciju logike upravljanja nekog procesa. Uslijed rasprostranjenosti korištenja Siemens tehnike TIA Portal ima široku primjenu u svijetu automatizacije industrijskih sustava, sa naglaskom korištenja za programiranje PLC uređaja i HMI sučelja.

Prije početka korištenja softverskog sučelja potrebno je osigurati konekciju između PLC-a i računala putem jednog od Ethernet standarda. Za ispravno povezivanje TIA Portala i PLC-a potrebno je poznavanje kataložnog broja i verzije CPU modula i sklopovlja, jer se programski podrazumijeva pretraživanje učitane baze gotovih Siemens proizvoda u procesu kreiranja rada u TIA Portalu [11].

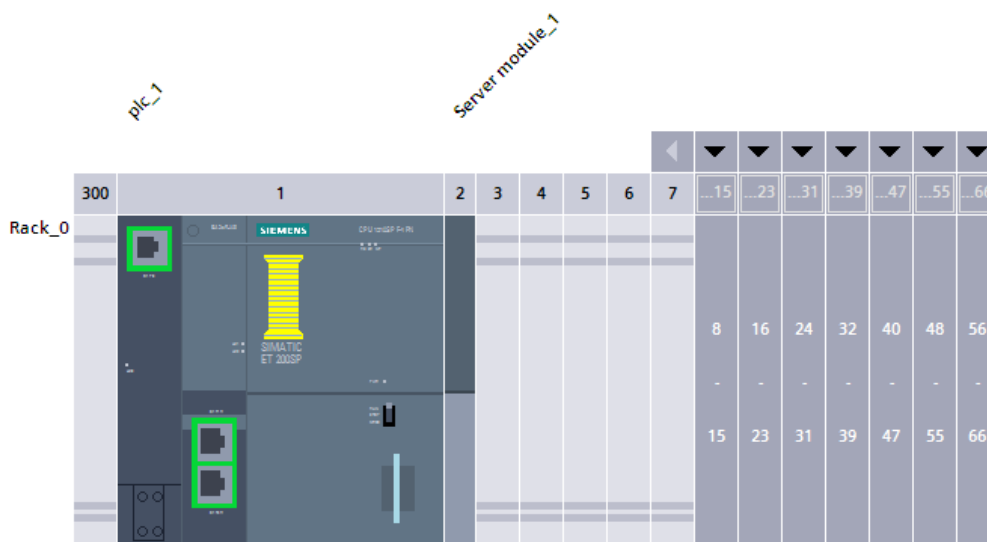
Razvoj samog programskog koda i logike upravljanja obavlja se direktno unutar softvera.

Pozivanje funkcijskih blokova (funkcija) podrazumijeva pretraživanje u bazi kataloških bojeva unutar TIA Portala u glavnom programu OB1. Nedostatak je potreba da se kreiraju pripadajući podatkovni blokovi (možda višekratno isti) čime se kreiraju njihove adrese u memoriji.



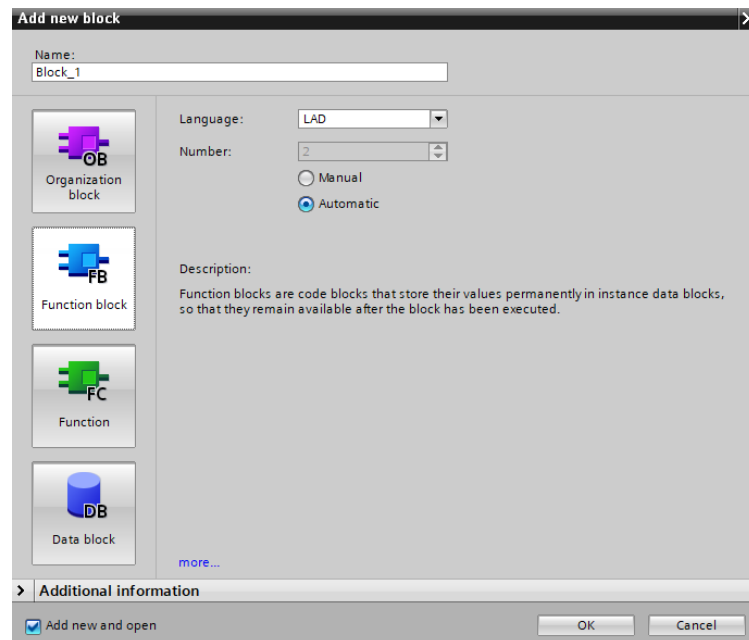
Slika 4.21 Odabir procesorske jedinice PLC-a

Ulazne varijable se upisuju pod *"Input"*, izlazne pod *"Output"*, ulazno - izlazne pod *"InOut"*, dok se privremene upisuju pod *"Temporary"* [11].



Slika 4.22 ET 200SP CPU prikazan unutar softvera TIA Portal

Odabrani CPU model biti će prikazan u "Device View" sekciji prikaza komponenata. Iz razloga što se u odabranom primjeru u radu koristi samo prednji X1 P3 priključak (gore lijevo na slici 4.22), isti će biti spojen i u softveru.



Slika 4.23 Dodavanje blokova

Nakon dodavanja ispravnog CPU dijela i odgovarajućih releja softver će automatski generirati organizacijski blok "Main". Moguće je dodati funkcije, funkcijske blokove ili podatkovne blokove. Funkcije sadržavaju lokalne varijable koje se izvršavaju samo prilikom poziva iste funkcije te nemaju alocirano memoriju za spremanje podataka, dok funkcijski blokovi spremaju memoriju unutar zasebno generiranih data blokova. Data (podatkovni) blokovi služe za spremanje globalnih ili lokalnih varijabli te poziv istih prilikom izvršenja funkcije. Na isti način potrebno je dodati HMI sučelje [11].

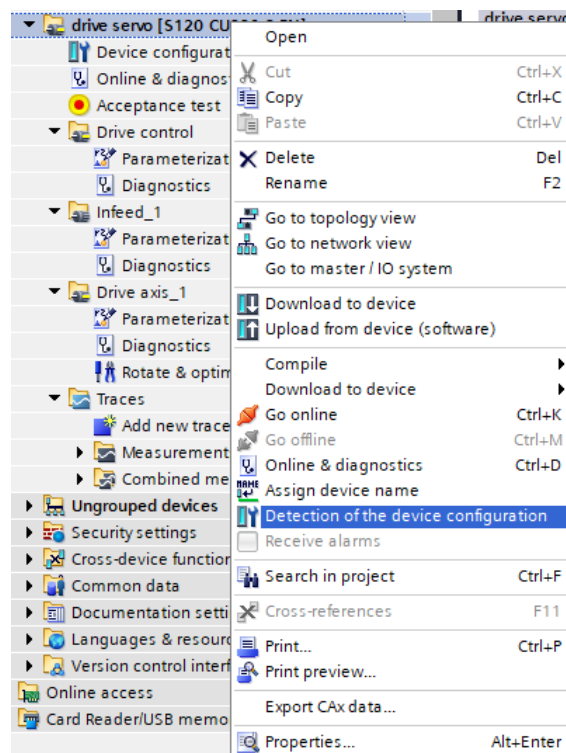
HMI sučelje omogućuje grafički i vizualni prikaz elemenata procesa automatizacije te pruža mogućnost upravljanja samim procesom na način da se definiraju HMI varijable sa odgovarajućim memorijskim bitovima koje su pritom spojene putem PLC-a na lokalne PLC varijable. Na taj način se sklopke, konektori, releji ili bilo koja druga vanjska fizikalna komponenta (koja bi se inače ručno morala kontrolirati) može upravljati putem HMI sučelja. Za ispravno povezivanje odabranog HMI sučelja sa PLC-om potrebno je poznavati komunikacijski driver koji se koristi te kataloški broj procesorske jedinice PLC-a.

Detaljnija razrada HMI sučelja biti će obrađena u narednim poglavljima.

Nakon što je dodan model procesorske jedinice PLC-a potrebno je dodati i frekvencijski pretvarač u program. Ovo je omogućeno na dva načina:

1. Ručno dodjeljivanje svake od komponenata frekvencijskog pretvarača na isti način kako je objašnjeno za procesorsku jedinicu PLC-a;
2. Preuzimanjem konfiguracije iz STARTER-a.

Radi jednostavnosti koristi se druga opcija. Programski je definiran Siemens S120 model frekvencijskog pretvarača te se koristi opcija detekcije konfiguracije pretvarača za automatsku uspostavu pretvarača.

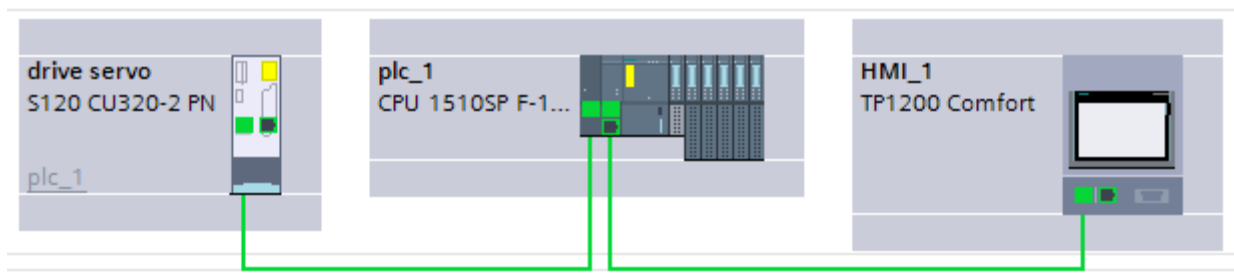


Slika 4.24 Detekcija konfiguracije pretvarača

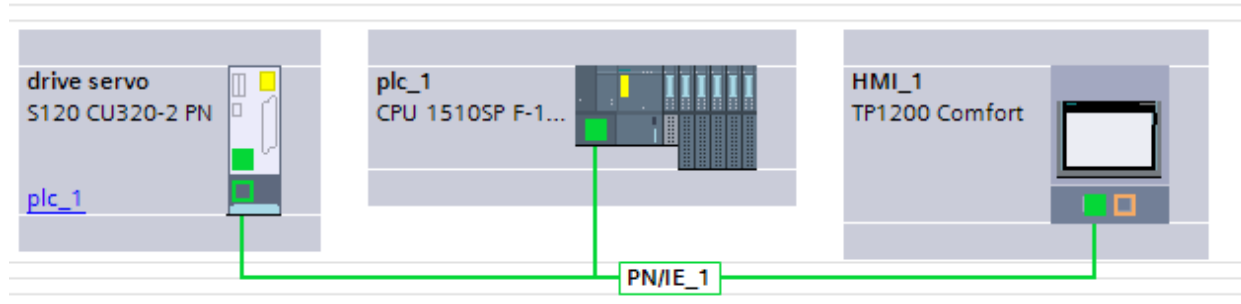
Softver automatski preuzima konfiguraciju objašnjenu u poglavlju 4.2. Generirani su sljedeći podsustavi:

- *Drive Control* – služi za parametrizaciju ulaza pretvarača i aktivaciju web servera,
- *Infeed\_1* – konfigurira parametre linijskog modula,
- *Drive axis\_1* – parametrizira pretvarač i omogućuje upravljanje motora putem korisničkog sučelja unutar TIA Portal-a,
- *Traces* – omogućuje grafički prikaz mjernih veličina.

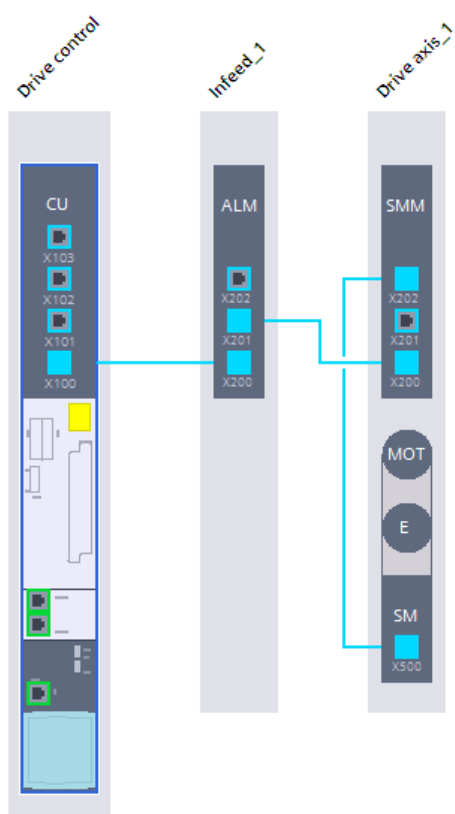
Nakon što su konfigurirani PLC i Drive potrebno je ispravno spojiti međusobne priključke unutar softvera na način na koji su spojeni prema slici 4.7.



Slika 4.24. Topološki prikaz konekcije svih navedenih podsustava (*Topology View*)



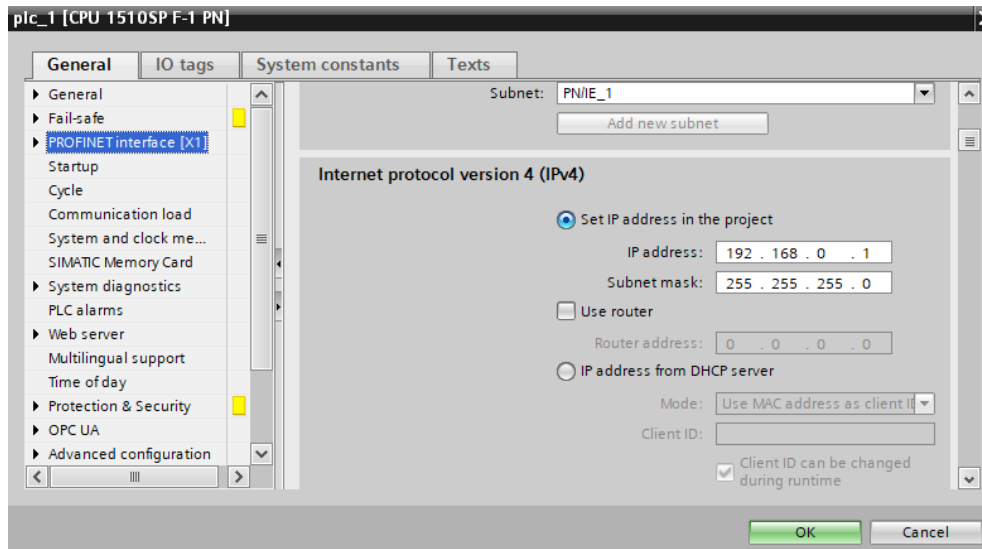
Slika 4.25 Mrežni prikaz konekcije svih navedenih podsustava (*Network View*)



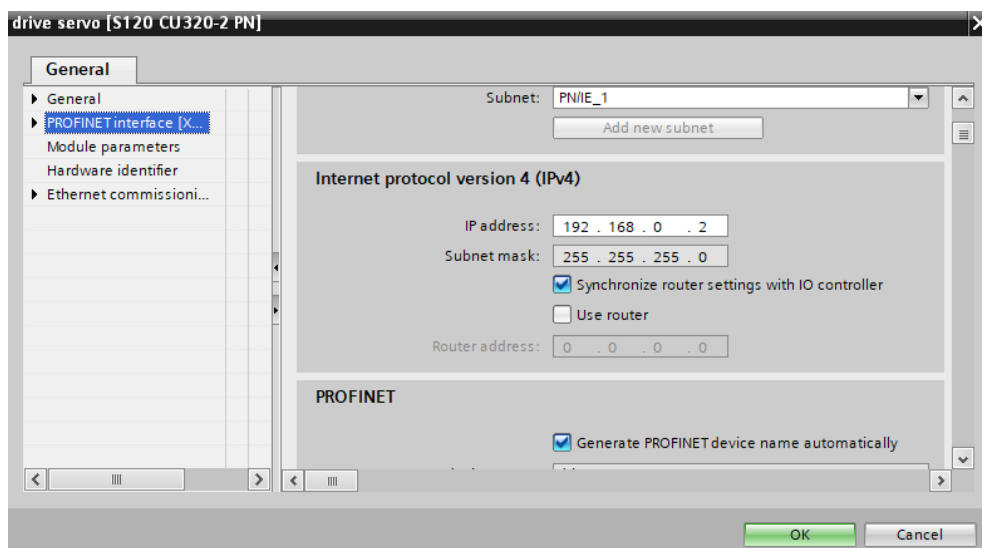
Slika 4.26 Konfiguracija pretvarača u TIA Portal-u

## 5. IZRADA PROGRAMSKE LOGIKE UPRAVLJANJA MOTOROM

Uspostava uspješne komunikacija između PLC-a i pretvarača podrazumijeva da su njihove IP adrese dijelovi istog subneta. Iz tog razloga IP adresa PLC uređaja u ovom radu postavljena je na vrijednost *192.168.0.1*, koja odgovara IP adresi S120 pretvarača.



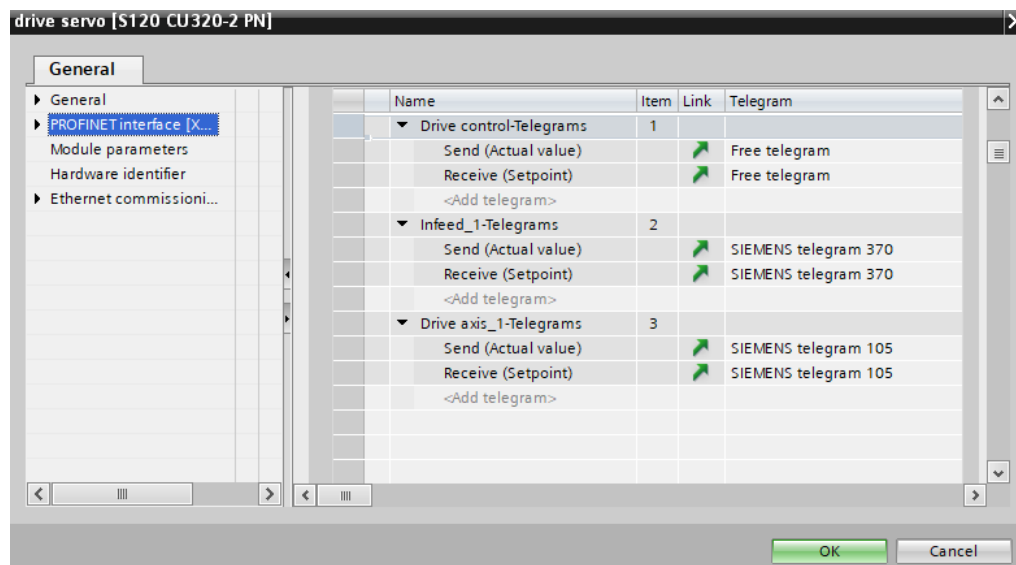
Slika 5.1 IPv4 adresa PLC uređaja



Slika 5.2 IPv4 adresa S120 pretvarača

Komunikaciju između PLC-a i Sinamics uređaja omogućuju komunikacijski protokoli zvani Telegrami koji služe za cikličku razmjenu podataka i informacijskih bitova. Postoje različite vrste Telegrama i diferenciraju se prema načinu upotrebe. Telegrami za upravljanje (Control Telegrams) se koriste se za slanje uputa i naredbi iz PLC-a prema Sinamics uređaju (pokretanje, zaustavljanje, promjenu brzine ili bilo koju drugu upravljačku naredbu).

Telegrami za povratne informacije (Feedback Telegrams) se koriste za prijenos povratnih informacija s Sinamics uređaja prema PLC-u (informacije o trenutnoj brzini, položaju ili bilo kojem drugom statusu uređaja). Telegrami za dijagnostiku (Diagnostic Telegrams) koriste se za prijenos dijagnostičkih informacija (praćenje statusa i detekciju eventualnih problema ili kvarova). Telegrami za konfiguraciju (Configuration Telegrams) se koriste se za postavljanje i konfiguraciju parametara Sinamics uređaja putem PLC-a.



Slika 5.3 Korišteni komunikacijski telegrami

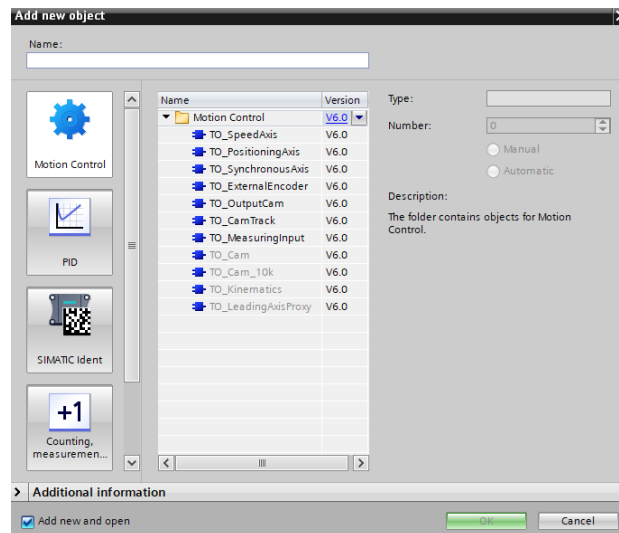
U diplomskom radu koriste se SIEMENS telegram 370 i 105 za komunikaciju komponenta frekvencijskog pretvarača sa PLC-om. Bitno je naglasiti kako je ispravan odabir komunikacijskog protokola neophodan iz razloga što različiti komunikacijski telegrami imaju različite upotrebe te krivi odabir komunikacijskog protokola rezultirao bi neispravnom komunikacijom međusobnih podsustava.

## 5.1. PLC program za upravljanje motorom

U svrhu omogućavanja lakšeg pregleda i efikasnijeg načina upravljanja kompleksnim i složenim sustavima poput asinkronih i sinkronih motora koristeći SIMATIC kontrolere uvodi se novi tip objekta u programsko okruženje zvan *"Technology Objects"*. Ovaj tip objekta služi za enkapsuliranje funkcija i lakši pristup parametrima upravljanih uređaja. Posebna podskupina tehnoloških objekata naziva se *"Motion Control"*, gdje postoje specifični tehnološki objekti koji služe za upravljanje i parametriranje osi motora. Tehnološki objekt za upravljanje osima (Axis Control Object) omogućava programiranje i kontrolu pojedinih osi motora (postavljanje ciljanih položaja, kontrola brzine, i upravljanje ubrzanjem/deceleracijom).

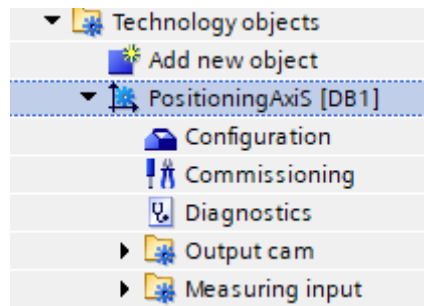
Tehnološki objekt za profiliranje brzine (Speed Profiling Object) omogućava postavljanje i praćenje profila brzine motora tijekom vremena u cilju postizanja glatkih i preciznih pokreta.

Tehnološki objekt za položaj (Positioning Object) omogućava postavljanje i praćenje ciljanih položaja motora i često koristi PID (Proporcionalni-Integralni-Derivativni) kontroler za preciznu kontrolu položaja.



Slika 5.4 Motion Control podskupina tehnoloških objekata

U svrhu upravljanja motora u diplomskom radu koristi se *"TO\_PositioningAxis"* tehnološki objekt prikazan na slici 5.4.



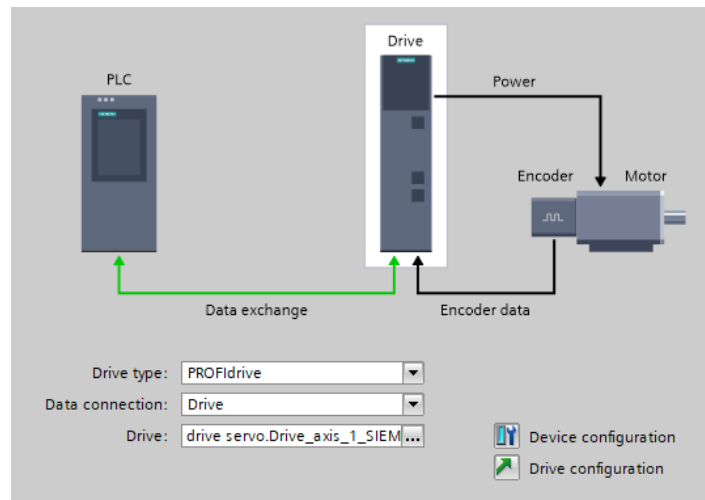
Slika 5.5 Positioning Axis

Tehnološki objekt *"TO\_PositioningAxis"* simulira mehaničku os motora. Ono omogućuje upravljanje pozicijom motora koristeći pozicijski setpoint. Nakon dodavanja tehnološkog objekta potrebno je ispravno konfigurirati hardversko sučelje, odnosno potrebno je konfigurirati sljedeće;

1. Drive
2. Enkoder
3. Razmjena podataka sa Drive-om
4. Razmjena podataka sa enkoderom

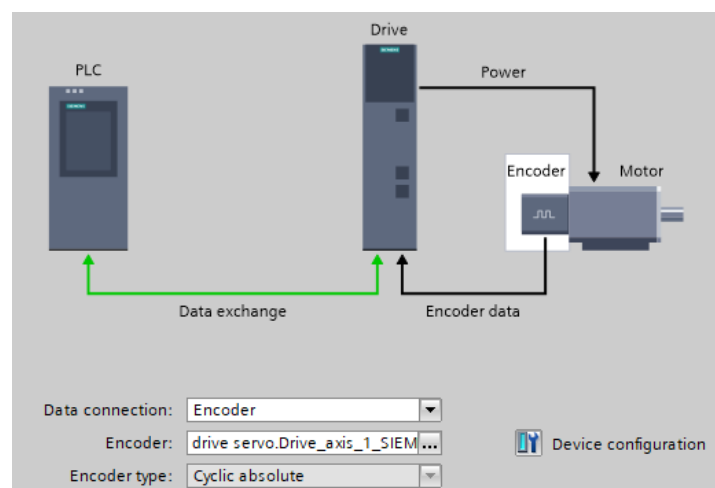


Slika 5.6 prikazuje konfiguracijsko sučelje Drive-a. Koristi se PROFI-Drive konekcija sa Drive-om i komunikacijski Telegram 105. Trenutno stanje pozicije osi motora je integrirano unutar tehnološkog objekta putem povratne veze te je omogućeno manipuliranje same osi motora.



Slika 5.6 Hardversko sučelje Drive-a

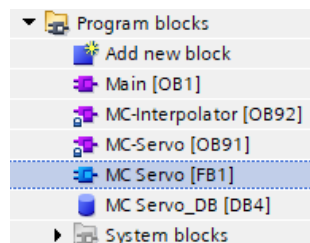
Konfiguracija apsolutnog cikličkog enkodera prikazana je na slici 5.7. Korišteni komunikacijski protokol je također Telegram 105. Putem komunikacijskog protokola enkoder ciklički šalje povratne informacije ka pretvaraču. Ovime je završena konfiguracija tehnološkog objekta i isti je spreman za pozivanje unutar PLC koda.



Slika 5.7 Hardversko sučelje enkodera

Ovakav postav omogućava operateru koristiti funkcionalnosti koje su dostupne kroz taj tehnološki objekt kako bi čitali informacije o položaju, brzini ili drugim relevantnim podacima iz enkodera, odnosno omogućava precizno praćenje položaja ili brzine motora koji koristi enkoder.

Za početak realizacije programskog koda za upravljanje sinkronim motorom kreiran je funkcijski blok nazvan "MC Servo". Kako je prethodno spomenuto, funkcijski blokovi prilikom poziva generiraju zasebne informacijske blokove ( "Data blokove" ) unutar kojih se nalaze globalne varijable koje mogu biti pozvane prilikom poziva same funkcije.



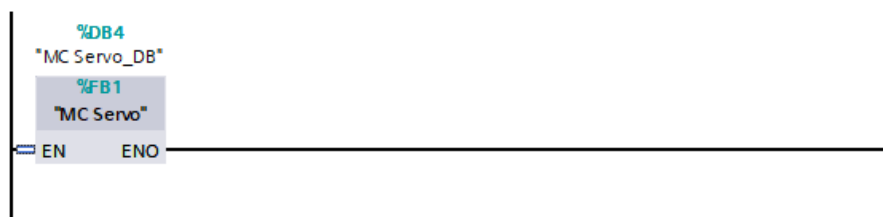
Slika 5.8 a) Funkcijski blok "MC Servo"

Ove varijable obično uključuju parametre konfiguracije, trenutne vrijednosti položaja i brzine motora, referentne točke, itd. (npr. maksimalna brzina, maksimalni položaj, kontrolne petlje za regulaciju brzine i položaja). TIA Portal će automatski generirati povezane informacijske blokove koji sadrže globalne varijable potrebne za rad tog specifičnog instance bloka.

MC Servo_DB								
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	InOut			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	MC_POWER_Instance	MC_POWER		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	MC_MOVERELATIVE_I...	MC_MOVERELATIVE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	MC_HALT_Instance	MC_HALT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	MC_MOVEVELOCITY_I...	MC_MOVEVELOCITY		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	MC_MOVERELATIVE_I...	MC_MOVERELATIVE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	MC_MOVEVELOCITY_I...	MC_MOVEVELOCITY		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	IEC_Timer_0_Instance...	TON_TIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	IEC_Timer_0_Instance...	TON_TIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	IEC_Timer_0_Instance...	TOF_TIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	IEC_Timer_0_Instance...	TON_TIME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

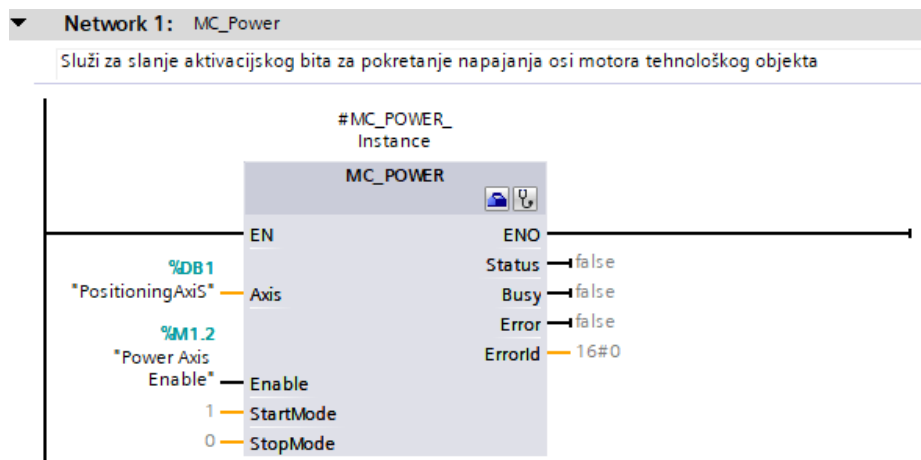
Slika 5.8 b) Varijable unutar data bloka "MC Servo"

Svaki funkcijski objekt, odnosno funkcijski blok mora biti sadržan unutar glavne "Main" funkcije koda. Prilikom pokretanja programa sve funkcije i globalne varijable pozivaju se istovremeno i program se izvršava.



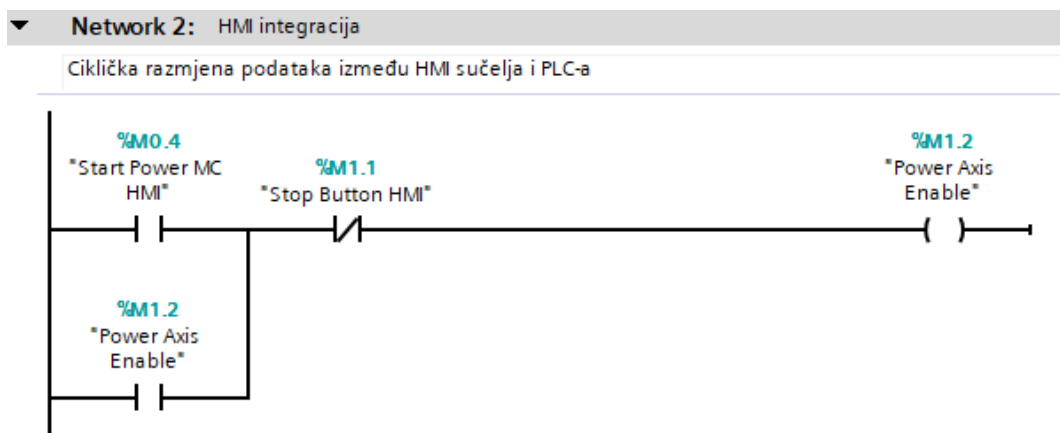
Slika 5.9 Poziv "MC\_Servo" data bloka unutar "Main" funkcije

Programski kod za pokretanje i upravljanje motora obavlja se unutar funkcijskog bloka "MC\_Servo". Dodan je "MC\_POWER" tehnološki objekt čija je glavna svrha slanja aktivacijskog bita za pokretanje osi motora.



Slika 5.10 Poziv tehnološkog objekta "MC\_Power"

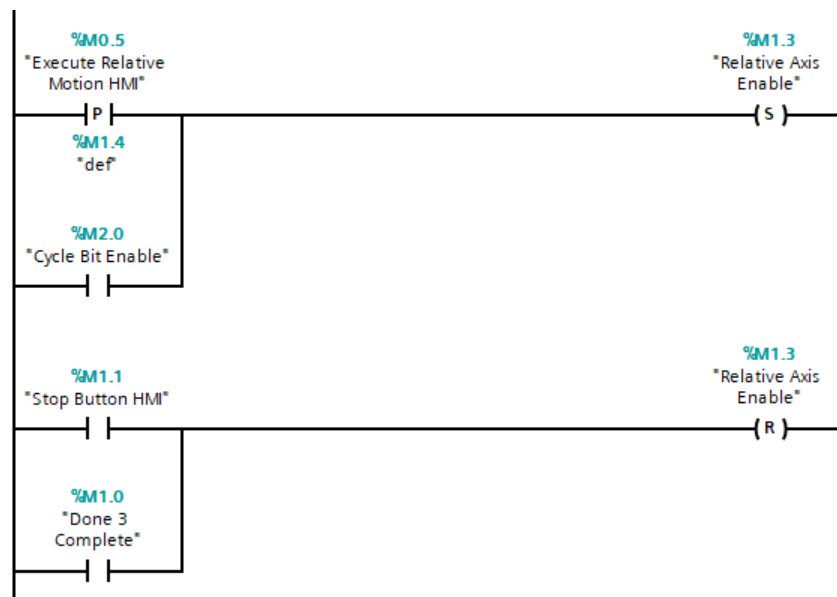
Tehnološki objekt poprima vrijednost pozicije osi motora na ulazu kao i signal za aktivaciju putem pomoćne memorijske varijable "Power Axis Enable" koji pokreće cikličko praćenje pozicije osi motora.



Slika 5.11 Pokretanje motora putem HMI sučelja

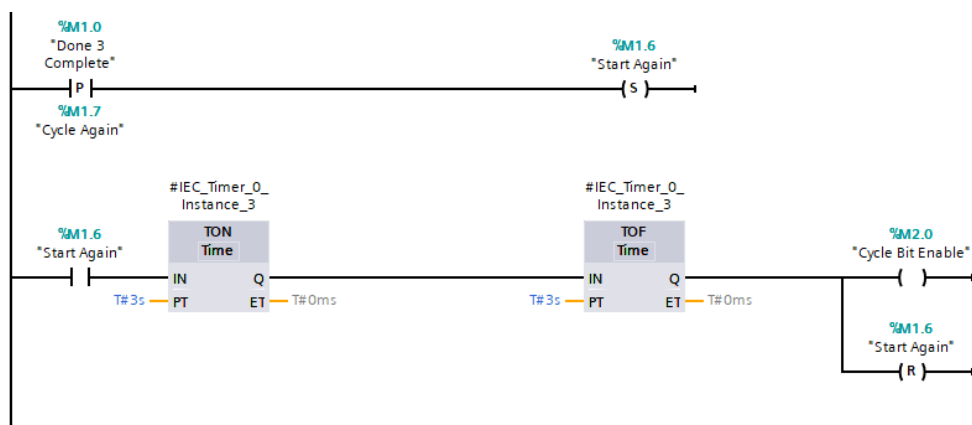
Memorijska varijabla "Start Power MC HMI" predstavlja tipkalo na HMI sučelju putem kojeg se pokreće motor, a na izlazu mreže postavljen je "Power Axis Enable" koji je dobio signal za aktivaciju. Ovaj način izvedbe predstavlja samouzбудu memorijskih varijabli, gdje prilikom prvog pokretanja motora pomoćne varijable trajno poprimaju vrijednost 1.

"Execute Relative Motion HMI" pomoćna varijabla (neki podatkovni element u programskom kodu koja vjerojatno sadrži informacije ili naredbe koje će se izvršiti kada dođe do određenog uvjeta) kao ulaz koristi "P\_TRIG" kontakt za detekciju uzlaznog brida signala (detektira uzlaznu tranziciju signala iz logičke 0 u logičku 1), nakon provjere memorijska varijabla "Relative Axis Enable" trajno poprima vrijednost 1 putem SET izlaza. Ovo se koristi u slučaju gdje određeni uvjeti moraju biti zadovoljeni prije odrađivanja specifične radnje.



Slika 5.12 Nastavak druge mreže

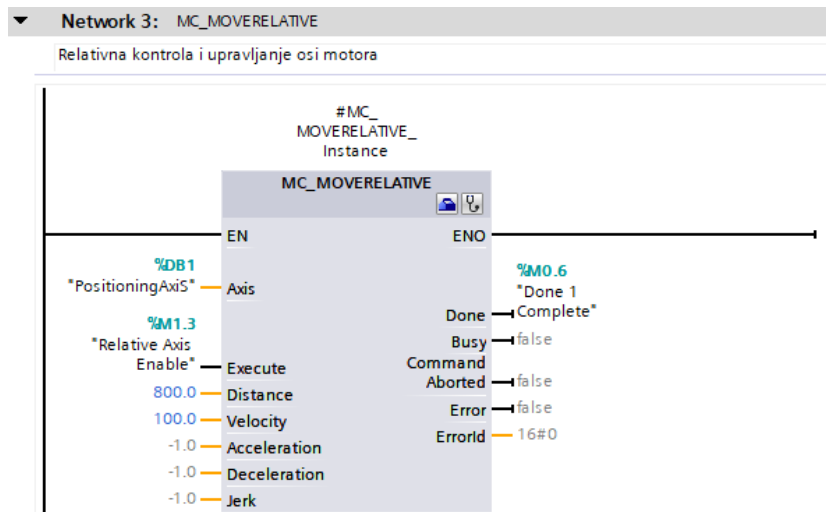
Na ovaj način se trajno prati trenutna vrijednost pozicije osi motora. Analogno tome zaustavljanjem procesa koristeći 'Stop Button HMI' se iznos prethodne varijable resetira na nulu.



Slika 5.13 Kraj druge mreže

Prilikom dovođenja aktivacijskog bita za ponovno pokretanje ciklusa rotacije osi motora aktivira se memorijska varijabla "Start Again". Uvjet za uključenje spomenute varijable jest taj da mora biti detektiran uzlazni brid "Done 3 Complete" varijable koja označava završetak originalnog ciklusa.

Nakon ponovnog pokretanja motora biti će izvršena dva Timer bloka prikazana na slici 5.13 označeni sa "*#IEC\_Timer\_0\_Instance\_3*". Prvi Timer blok je tipa TON, koji nakon predefiniiranog vremena kašnjenja (u ovom slučaju tri sekunde) postavlja izlaz na 1 i signal za aktivaciju poprma Timer blok TOF. Vrijedi ista analogija kao za prethodni slučaj, jedina razlika je u tome što nakon isteka predefiniiranog vremena kašnjenja (ponovno tri sekunde) resetira se "*Start Again*" memorijska varijabla i signal za aktivaciju poprma ciklički bit koji šalje instrukciju za ponovno pokretanje varijable "*Relative Axis Enable*".

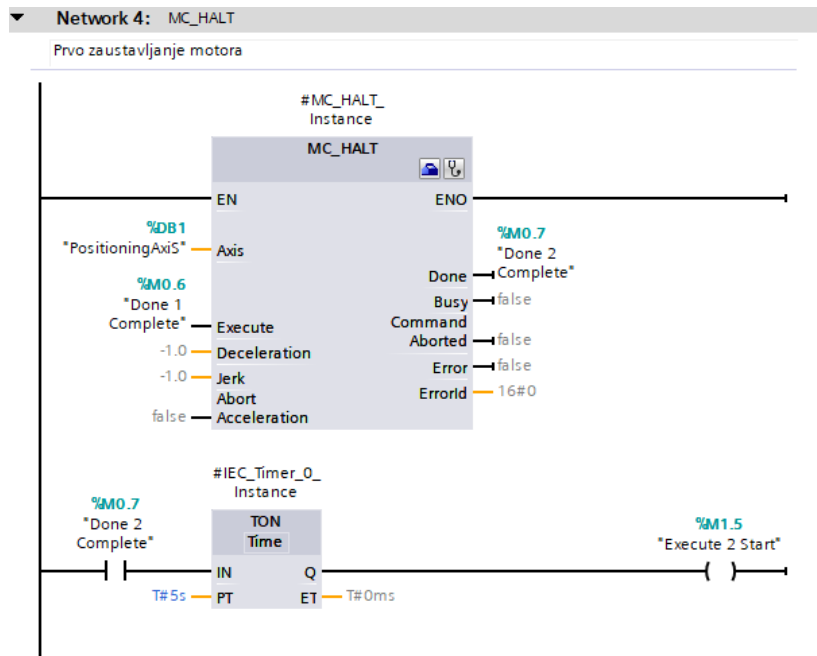


Slika 5.14 Poziv tehnološkog objekta "*MC\_Moverelative*"

U sljedećoj mreži poziva se novi tehnološki objekt "*MC\_Moverelative*". Pri tome skraćenice definiraju MC (Motion Control) kao objekt povezan s upravljanjem pokretom (uključuje kontrolu brzine, položaja, ubrzanja i sl.), dok oznaka *Move Relative* sugerira da je tehnološki objekt namijenjen relativnim pokretima (postavljanje i izvođenje pokreta s obzirom na trenutni položaj motora, umjesto apsolutnih pozicija).

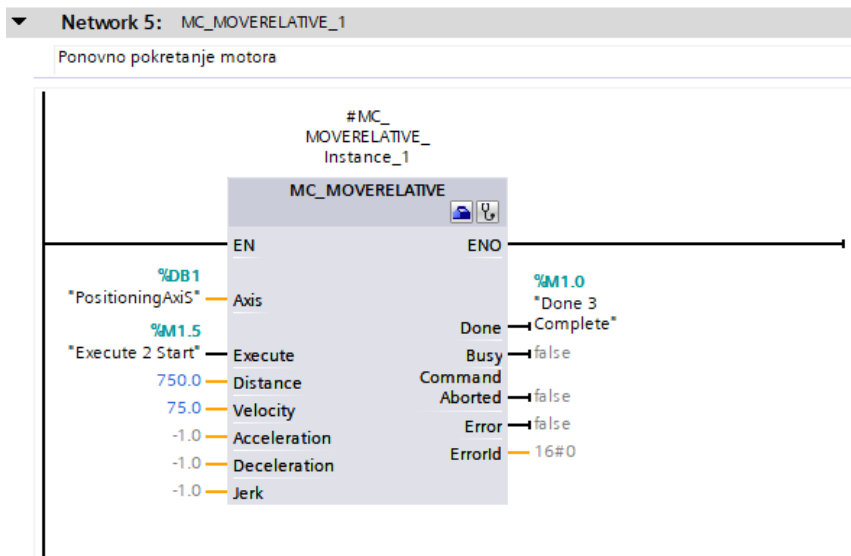
Definirana su četiri ulaza – prvi ulaz poziva "*PositioningAxis*" tehnološki objekt unutar kojeg je spremljena trenutna vrijednost pozicije osi motora dok drugi ulaz poziva "*Relative Axis Enable*" koji šalje bit za signalizaciju aktivacije praćenja promjene pozicije osi motora detektiranjem uzlaznog brida signala. Izlaz tehnološkog objekta spremljen je u memorijskoj varijabli "*Done 1 Complete*". Nakon estimacije trenutnog položaja osi motora naspram promjene položaja osi motora informacija se skladišti u memorijski registar procesorske jedinice PLC-a. U tehnološkom objektu dodatna dva ulaza su "*Distance*" i "*Velocity*" parametri, odnosno ulazi predstavljaju odnos tražene pozicije naspram relativne pozicije. Na primjer, "*Distance*" može biti apsolutna pozicija koja se treba postići, dok "*Velocity*" može odrediti brzinu kojom je potrebno da se motor kreće prema toj poziciji.

Tehnološki objekt "MC\_Halt" zaustavlja os motora koristeći rampu za kočenje. Ulaz je prethodno spomenuti tehnološki objekt "PositioningAxis" koji prati iznos pozicije osi motora i memorijska varijabla "Done 1 Complete" koja signalizira zaustavljanje ciklusa detekcijom uzlaznog brida na ulazu.



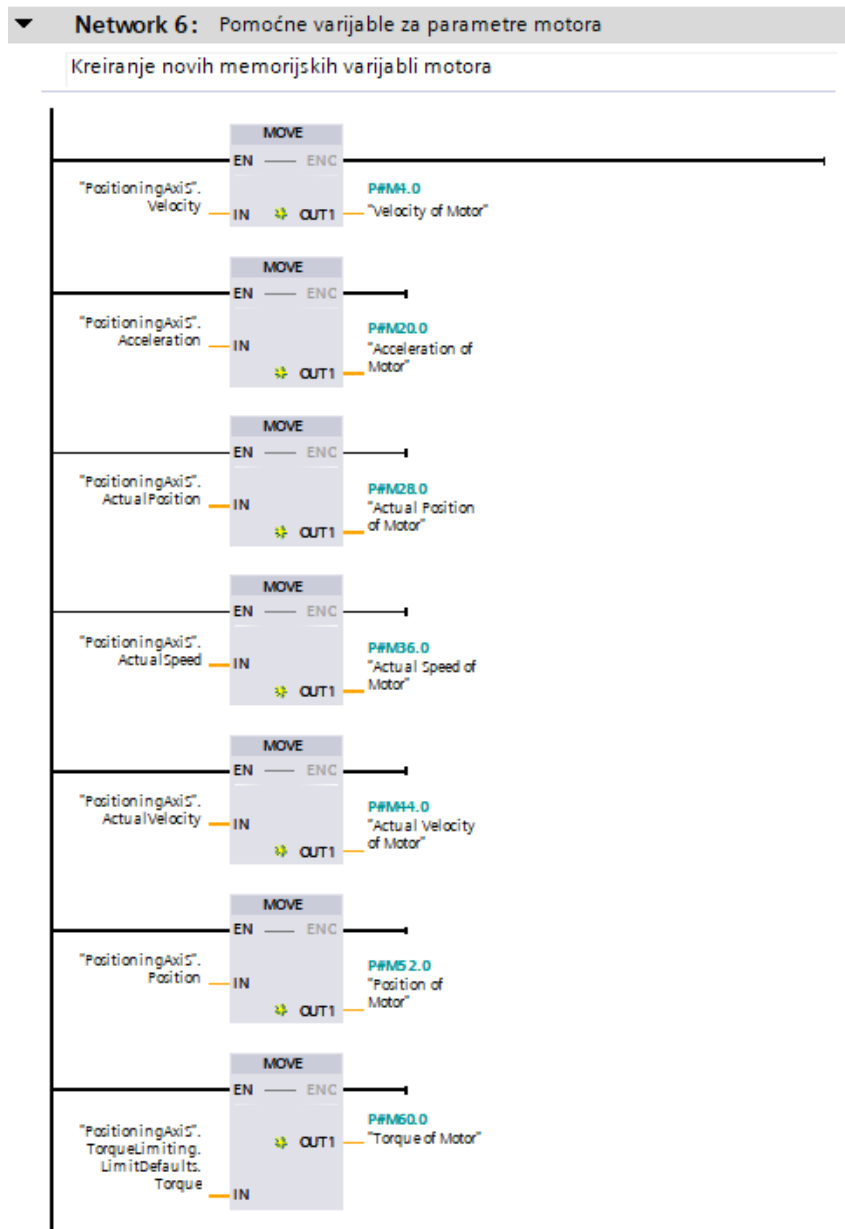
Slika 5.15 Poziv tehnološkog objekta "MC\_Halt"

Izlaz tehnološkog objekta je memorijska varijabla "Done 2 Complete" koja označava kraj "MC\_Halt" funkcije te prilikom postizanja signala za aktivaciju uključuje se TON Timer u trajanju 5 sekundi, nakon čega se aktivira "Execute 2 Start" – zaustavljajući motor.



Slika 5.16 Drugi poziv "MC\_Moverelative" tehnološkog objekta

Ista analogija kako je prethodno objašnjena u primjeru prikazanom na slici 5.14 vrijedi i na slici 5.16, jedina razlika je što je izlaz tehnološkog objekta spremljen u memorijskoj varijabli "Done 3 Complete" i ciklus kompletno završava te počinje ispočetka kada se na ulazu spomenute varijable detektira uzlazni brid (slika 5.13).



Slika 5.17 Spremanje parametara motora u nove varijable

U zadnjoj mreži svi potrebni parametri motora spremaju se u novo kreirane varijable. Ove varijable uključuju brzinu rotacije motora, moment motora, poziciju motora, akceleraciju i brzinu motora.

Svaka varijabla koja se koristi prilikom izvođenja logike upravljanja PLC-a mora biti skladištena unutar zasebne tablice ("*Tag table*"). Ove tablice služe kao skupovi varijabli različitih tipova podataka i omogućuju strukturiranje i pristup podacima unutar PLC programa. Tip podataka ovih varijabli može biti različit, primjerice Bool, Real, LReal itd. ovisno što se pohranjuje u memorijsku adresu varijable. Ulazi imaju predefiniciranu sintaksu memorijske adrese sa "*%IX.Y*", izlazi su "*%QX.Y*" a memorijske varijable su prethodno spomenute kao "*%MX.Y*".

PLC_tags							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
	PositioningAxis_Actor_Interfac...	"PD_TEL105_IN"	%I260.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	PositioningAxis_Actor_Interfac...	"PD_TEL105_O..."	%Q260.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	MC_Power_PB	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	MC_RST_PB	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	MC_FWR_PB	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	MC_REV_PB	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Start Power MC HMI	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Execute Relative Motion HMI	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Done 1 Complete	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Done 2 Complete	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Done 3 Complete	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Stop Button HMI	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Power Axis Enable	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Relative Axis Enable	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	def	Bool	%M1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Execute 2 Start	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Start Again	Bool	%M1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Cycle Again	Bool	%M1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Cycle Bit Enable	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Velocity of Motor	LReal	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Velocity of Acceleration	LReal	%M12.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Acceleration of Motor	LReal	%M20.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Actual Position of Motor	LReal	%M28.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Actual Speed of Motor	LReal	%M36.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Actual Velocity of Motor	LReal	%M44.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Position of Motor	LReal	%M52.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Torque of Motor	LReal	%M60.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Slika 5.18 Sve korištene varijable PLC programa

Putem "*Trace*" sučelja moguće je odabrati koji signali će grafički biti prikazani. U ovom slučaju odabrana su četiri signala prikazana na slici 5.19. Nakon izvršenja programske logike upravljanja motorom potrebno je izraditi pripadajuće HMI sučelje.

Signals			
	Name	Data type	Address
1	"Acceleration of Motor"	LReal	P#M20.0
2	"Actual Speed of Motor"	LReal	P#M36.0
3	"Actual Position of Motor"	LReal	P#M28.0
4	"Torque of Motor"	LReal	P#M60.0
5			<Add>

Slika 5.19 Izabrane varijable za grafički prikaz



## 5.2. HMI realizacija

Implementacija HMI (Human-Machine Interface) sučelja vrlo je slična konfiguraciji PLC uređaja kako je objašnjeno u poglavlju 4.4. Uobičajeno korištenje HMI sučelje podrazumijeva vizualizaciju i praćenje odabranih signala. Osnovni koraci u tom procesu su: 1) definiranje HMI projekat, 2) dodavanje grafičkih elemenata (grafička polja, kontrole,...), 3) povezivanje signala koje pratite putem "Trace" sučelja s odgovarajućim grafičkim elementima, 4) konfiguriranje animacija i trendova (bolji vizualni dojam i praćenje signala), 5) postavke alarma za praćenje određenih uvjeta ili događaja. Konačno kreirano HMI sučelje potrebno je testirati, optimirati i korigirati, te konačno pohraniti.

HMI sučelje ima zasebne tablice isto kao i PLC, a varijable koje HMI koristi moraju biti direktno povezane sa varijablama PLC-a. Odabran je TP1200 Comfort Panel HMI sučelje. Nakon dodavanja HMI sučelja potrebno je ostvariti konekciju sa PLC uređajem na način prikazan na slici 5.20.

Connections							
Name	Communication driver	HMI time synchronization mode	Station	Partner	Node	Online	
HMI_Connection_1	SIMATIC S7 1500	None	ET.200SP station_1	plc_1	CPU 1510SP F-1 PN...	<input checked="" type="checkbox"/>	
<Add new>							

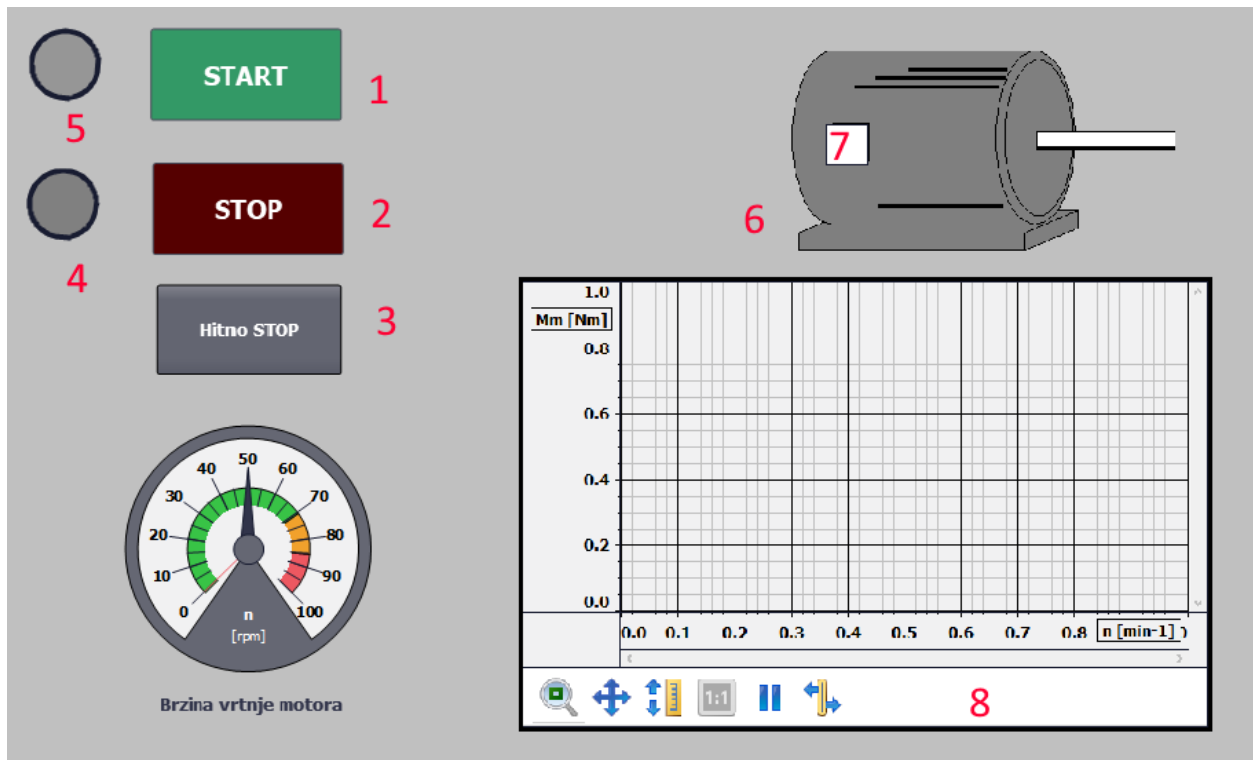
Slika 5.20 Povezivanje HMI sučelja sa procesorskom jedinicom PLC-a

Svaka PLC varijabla prikazana na slici 5.21 povezana je sa odgovarajućom HMI varijablom. Prilikom izvođenja simulacije putem HMI sučelja koristit će se HMI varijable.

HMI_tablica					
Name	Data type	Connection	PLC name	PLC tag	
MC_Power_PB	Bool	HMI_Connection...	plc_1	MC_Power_PB	
MC_RST_PB	Bool	HMI_Connection...	plc_1	MC_RST_PB	
MC_FWR_PB	Bool	HMI_Connection...	plc_1	MC_FWR_PB	
MC_REV_PB	Bool	HMI_Connection...	plc_1	MC_REV_PB	
Start Power MC HMI	Bool	HMI_Connection...	plc_1	*Start Power MC HMI*	
Execute Relative Motion HMI	Bool	HMI_Connection...	plc_1	*Execute Relative Motion ...	
Done 1 Complete	Bool	HMI_Connection...	plc_1	*Done 1 Complete*	
Done 2 Complete	Bool	HMI_Connection...	plc_1	*Done 2 Complete*	
Done 3 Complete	Bool	HMI_Connection...	plc_1	*Done 3 Complete*	
Stop Button HMI	Bool	HMI_Connection...	plc_1	*Stop Button HMI*	
Position Axis Enable	Bool	HMI_Connection...	plc_1	*Power Axis Enable*	
Relative Axis Enable	Bool	HMI_Connection...	plc_1	*Relative Axis Enable*	
def	Bool	HMI_Connection...	plc_1	def	
Execute 2 Start	Bool	HMI_Connection...	plc_1	*Execute 2 Start*	
Start Again	Bool	HMI_Connection...	plc_1	*Start Again*	
Cycle Again	Bool	HMI_Connection...	plc_1	*Cycle Again*	
Cycle Bit Enable	Bool	HMI_Connection...	plc_1	*Cycle Bit Enable*	
Velocity of Motor	LReal	HMI_Connection...	plc_1	*Velocity of Motor*	
Velocity of Acceleration	LReal	HMI_Connection...	plc_1	*Velocity of Acceleration*	
Acceleration of Motor	LReal	HMI_Connection...	plc_1	*Acceleration of Motor*	
Actual Position of Motor	LReal	HMI_Connection...	plc_1	*Actual Position of Motor*	
Actual Speed of Motor	LReal	HMI_Connection...	plc_1	*Actual Speed of Motor*	
Actual Velocity of Motor	LReal	HMI_Connection...	plc_1	*Actual Velocity of Motor*	
Position of Motor	LReal	HMI_Connection...	plc_1	*Position of Motor*	
Torque of Motor	LReal	HMI_Connection...	plc_1	*Torque of Motor*	

Slika 5.21 Varijable HMI sučelja povezane sa varijablama PLC-a

Nakon uspješnog uspostavljanja veze između PLC uređaja i HMI sučelja te deklariranje svih potrebnih varijabli moguće je dizajnirati izgled HMI sučelja.



Slika 5.22 HMI sučelje

Prilikom dodavanja HMI sučelja u konfiguraciju PLC-a automatski je dodijeljen "root screen" na kojemu se vrši sav grafički prikaz prikazan na slici 5.22. Sučelje realizirano u ovom radu se sastoji od sljedećih dijelova;

1. Tipkalo za pokretanje motora,
2. Tipkalo za zaustavljanje motora,
3. Sklopka za hitno gašenje motora,
4. LED signalizacija aktivacije motora,
5. LED signalizacija deaktivacije motora,
6. Simbol motora,
7. LED signalizacija rada motora,
8. Grafički prikaz momentne karakteristike motora.

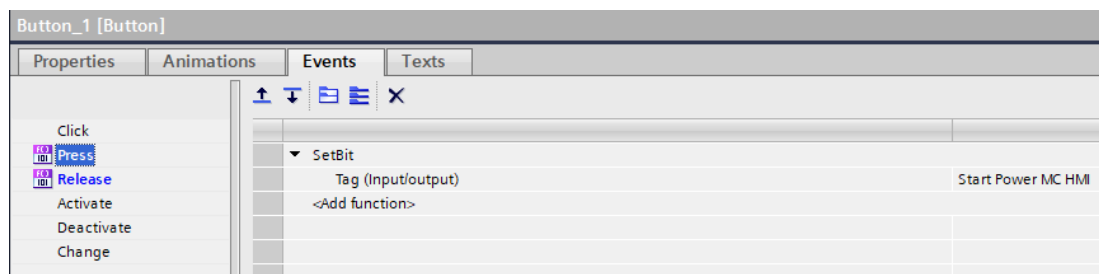
Svaki od prikazanih elemenata HMI sučelja ima dodijeljenu odgovarajuću HMI varijablu.

Primjerice, tipkalo START ima dodijeljeno varijablu "Start Power MC HMI" dok STOP tipkalo ima dodijeljeno varijablu "Stop Power MC HMI".

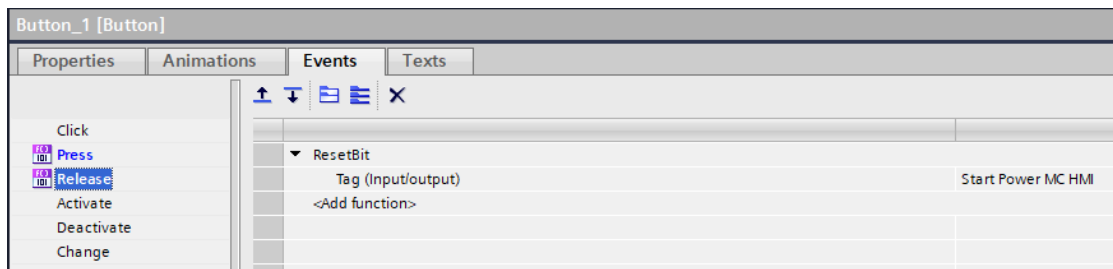
Prilikom korištenja tipkala unutar HMI sučelja potrebno je dodijeliti funkcije koje će tipkalo obavljati nakon što je pritisnuto. Pritisak tipkala vrši se u dvije faze – pritisak tipkala te puštanje tipkala. Prilikom pritiska tipkala poziva se funkcija "SetBit" koja mijenja vrijednost ulaznog bita varijable sa nule na jedinicu. Ovo vrijedi samo dokle se tipkalo drži u pritisnutom stanju. Puštanjem tipkala poziva se funkcija "ResetBit" koja tipkalo vraća u originalno stanje, odnosno ulazni bit se resetira na vrijednost nulu. Tipkalo također mora biti povezano sa varijablom koja se koristi prilikom izvođenja PLC koda.

Tipkalo "START" povezano je PLC varijablom "Start Power MC HMI" te aktivacija iste varijable putem HMI sučelja uzrokuje postavljanje vrijednosti ulaznog bita te varijable na jedinicu, nakon čega se pokreće proces samouzbude varijabli i pokretanja motora kako je prikazano na slici 5.11.

Tipkalo "STOP" povezano je sa PLC varijablom "Stop Button HMI" te vrijedi ista analogija kako je objašnjena za prethodno tipkalo. Sklopka "Hitno STOP" služi za automatsko zaustavljanje sekvence izvođenja logike upravljanja motorom na način da se postavi zatvoreni kontakt na ulaz "Power Axis Enable" varijable, odnosno pokretanje sklopke uzrokuje promjenu stanja kontakta na otvoren kontakt te se kompletan proces pokretanja motora zaustavlja. Ponovnim pritiskom na sklopku reaktivira se proces i moguće je ponovno pokrenuti motor. Ova sklopka uvedena je kao sigurnosna mjera ukoliko se pojavi potreba za hitnim isključenjem motora.

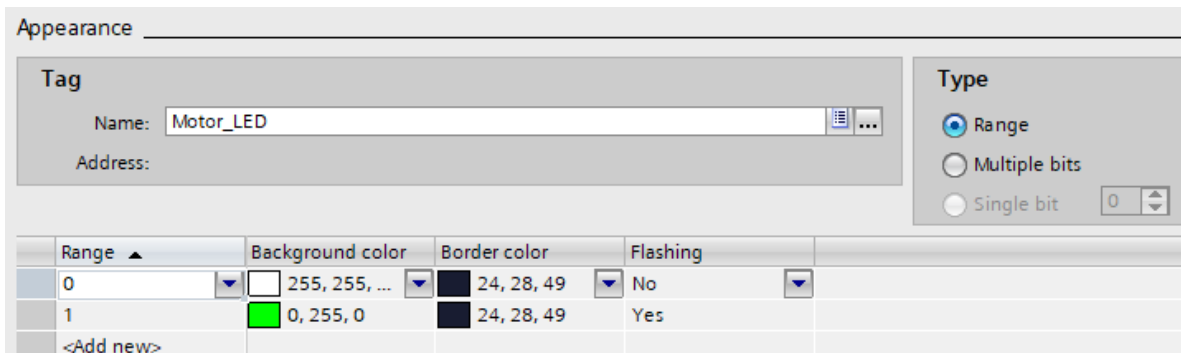


Slika 5.23 "SetBit" funkcija za varijablu "Start Power MC HMI"



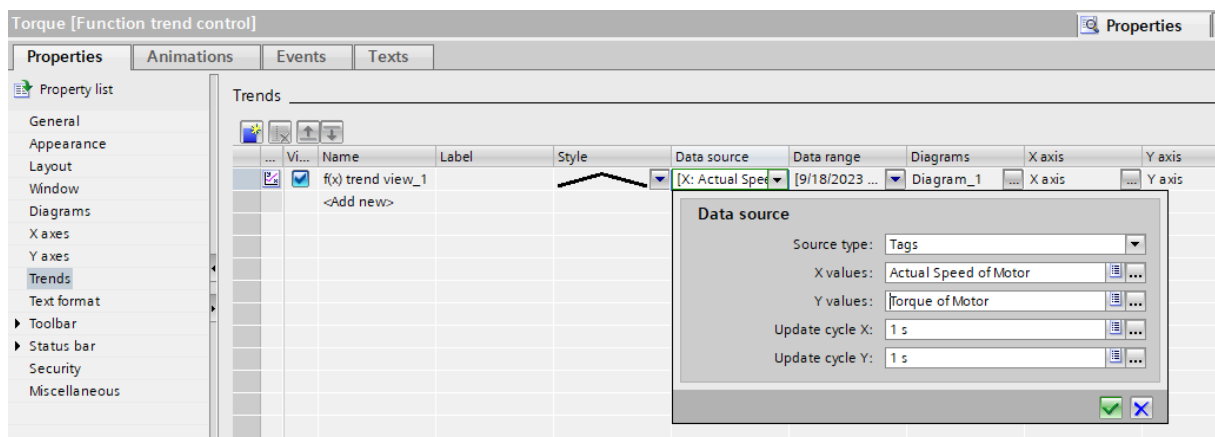
Slika 5.24 "ResetBit" funkcija za varijablu "Start Power MC HMI"

Na slici 5.22 pod oznakom 6 prikazan je simbol motora koji služi kao grafičku reprezentaciju stanja motora. U HMI sučelje dodana je posebna LED rasvjeta za signalizaciju rada motora. Aktivacija LED rasvjete unutar HMI sučelja izvodi se na način da se objektu dodaju zasebne animacije unutar "Animations" sučelja u postavkama samog objekta. Svaki objekt koji služi za svjetlosnu signalizaciju ima postavljen rang bitova od nule do jedinice, gdje nakon što je detektirana jedinica na ulazu aktivira se postavljena animacija.



Slika 5.25 Postavljanje svjetlosne signalizacije rada motora

Pod brojem 8 na slici 5.22 prikazano je grafičko sučelje za prikaz momentne karakteristike motora. Prilikom definiranja ulaznih varijabli apscise i ordinate unesene su sljedeće varijable:



Slika 5.26 Konfiguracija momentne karakteristike

Number	Parameter text	Value	Unit
p2000	Reference speed	3000.00	rpm
p2001	Reference voltage	1000	Vrms
p2002	Reference current	15.00	Arms
p2003	Reference torque	19.95	Nm
r2004	Reference power	6.27	kW
p2005	Reference angle	90.00	°
p2006	Reference temperature	100.00	°C
p2007	Reference acceleration	50.00	rev/s <sup>2</sup>

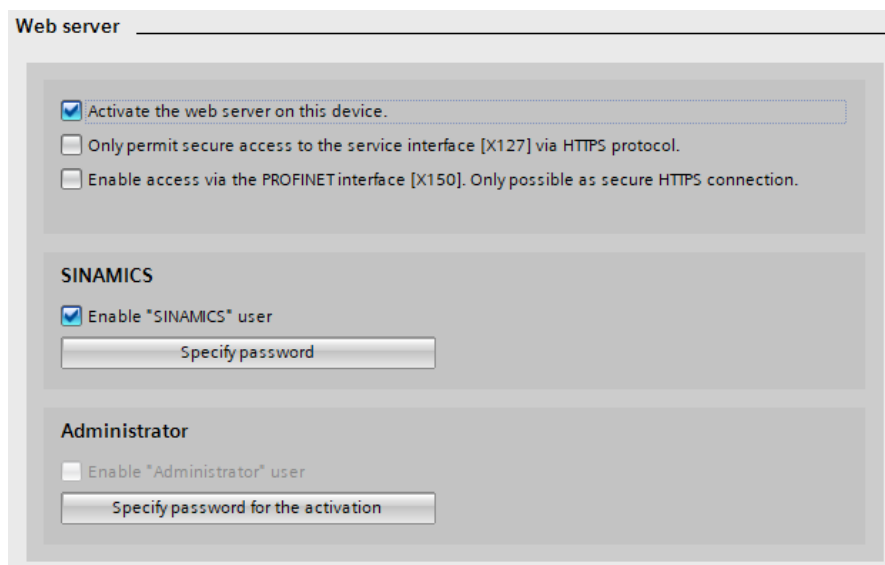
Slika 5.27 Parametri motora

## 6. WEB SUČELJE

Frekvencijski pretvarač Siemens Sinamics S120 u sebi ima ugrađen *Firmware* koji omogućuje korisniku puštanje u pogon i konfiguraciju samog frekvencijskog pretvarača putem interaktivnog web sučelja. Prema blokovskom dijagramu prikazanom na slici 4.7 frekvencijski pretvarač spojen je sa računalom kroz Ethernet sklopku koristeći priključak X150 P1. IP adresa ovog priključka postavljena je na iznos *192.168.0.2* što znači da se na toj adresi odvija sva komunikacija i razmjena podataka između pretvarača i računala putem PROFINET veze. Frekvencijski pretvarač koristi programski jezik MWSL (engl. "*Mini Web Server Language*") za obradu parametara motora i konfiguraciju web sučelja.

### 6.1. Konfiguracija web sučelja

Za uspostavu web servera pretvarača potrebno je omogućiti opciju aktivacije servera putem STARTER softvera ili unutar konfiguracije parametara pretvarača koristeći TIA Portal.

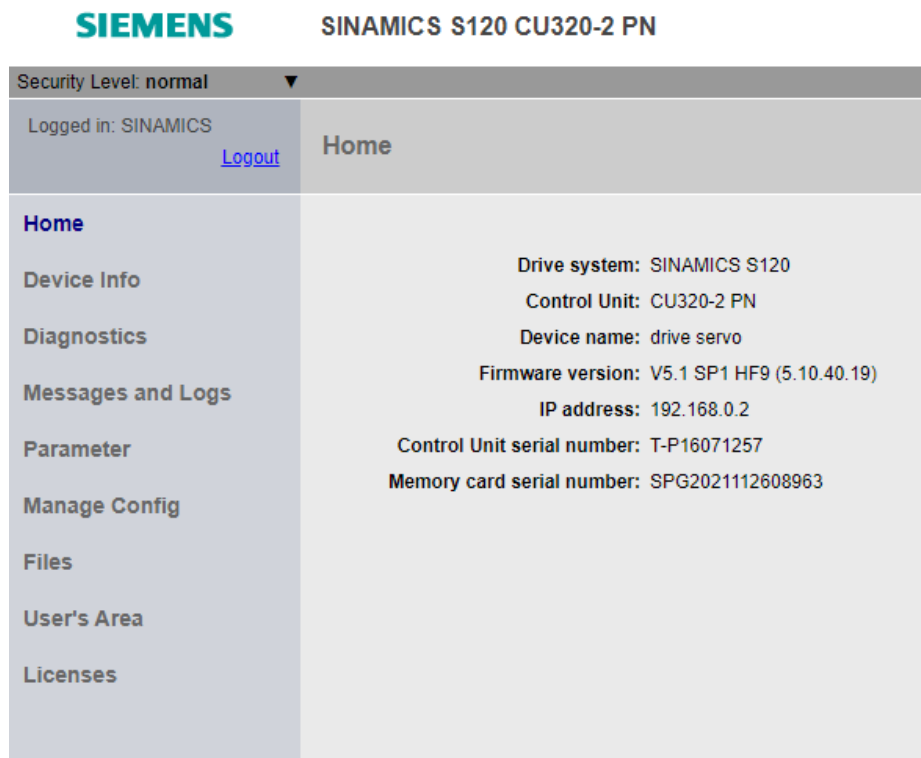


Slika 6.1 Aktivacija web servera

Pri tvorničkim postavkama omogućen je pristup web serveru kao "SINAMICS" korisnik, no za upravljanje konfiguracije web sučelja potreban je administratorski pristup. Moguće je postaviti lozinku putem koje će se korisnik ulogirati u sučelje web servera kao administrator.

Nakon što je omogućen pristup web serveru realiziran je HTTP protokol koji omogućuje lokalno spajanje web servera sa IP adresom pretvarača. Pristup web serveru moguć je kroz jedan od internetskih pretraživača (npr. Google Chrome, Microsoft Edge, Internet Explorer). Upisuje se IP adresa u pretraživač te se otvara uvodna stranica web servera.

Na uvodnoj stranici web servera (slika 6.2) nalaze se osnovni podaci frekvencijskog pretvarača poput serijskog modela upravljačke jedinice pretvarača, naziv pretvarača korišten u PLC konfiguraciji, IP adresa frekvencijskog pretvarača putem kojeg se korisnik spojio na web server te serijski broj memorijske kartice pretvarača.



Slika 6.2 Uvodna stranica web servera

Ukoliko korisnik nije ulogiran određeni parametri frekvencijskog pretvarača neće biti prikazani te daljnja konfiguracija web servera neće biti moguća. U slučaju da je korisnik ulogiran u "SINAMICS" račun po tvorničkim postavkama samo će pregled web servera biti omogućen. Za kompletni pristup svim aspektima web servera potreban je administratorski pristup.

Component	Identification	Status	No.	FW version	Type	Order No.	HW	Serial No.	Own Port	Uplink to (Port, No.)	
Drive control drive servo	Identification via LED	■	1	05.10.40.19	CU320-2 PN	8SL3040-1MA01-0AA0	F	ST-P16071257	0	0	0
Infeed_1 Line Module_1	Identification via LED	■	2	05.10.40.05	LM_ACDC	8SL3130-7TE21-6AA4	C	T-ND6192305	0	0	1
Drive axis_1 Motor Module_1	Identification via LED	■	3	05.10.40.05	MM_1AXIS_DCAC	8SL3120-1TE15-0A00	B	T-N76237767	0	1	2
Drive axis_1 Encoder evaluation_1	Not possible	■	6	02.01.09.06	SIM20 / DIQI	8FX2001-5JE20-2DC0	G	A2089142448	0	0	5

**Entry** ■ Component exists in the actual topology, but not in the target topology  
**Entry** ■ Component exists in the target topology, but not in the actual topology  
**Status** ■ Ready  
**Status** ■ Fault  
**Status** ■ Error




Slika 6.3 Informacije uređaja

Dijelovi frekvencijskog pretvarača prikazane su u odjeljku "*Device Info*" web servera. Na slici 6.3 evidentna je ista konfiguracija dijelova pretvarača na način kako je prikazano 4.24. Konfiguracija se sastoji od sljedećih komponenti;

1. Drive Control
2. Linijski modul
3. Motorni modul
4. Evaluacija enkodera

Radi provjere uspješne konekcije između komponenata pretvarača sa web serverom moguće je odabrati opciju "*Identification via LED*". Pritiskom tipkala LED diode komponenata pretvarača počinju treperiti čime je signalizirana uspješna povezanost između web servera i pretvarača. U informacijama uređaja također je prikazana firmware verzija svake komponente, njihovi zasebni nazivi i serijski brojevi.

Dijagnostički prikaz na slici 6.4 prikazuje stanje komponenata frekvencijskog pretvarača te javlja korisniku ukoliko se pojavio kvar ili je postavljena kriva konfiguracija komponenata. U slučaju da je detektirana neispravna konfiguracija softver automatski nudi rješenje pod "*Operational display*" prikazom.

Diagnostics - Service overview				
Service overview		Trace files		
DO	DO name	DO type / view	Faults/alarms	Operational display (r2)
1	Drive control	SINAMICS S		Ready(10)
2	Infeed_1	ACTIVE INFEED CONTROL		Switching on inhibited - set "OC/OFF2" = "1" (p0844, p0845)(42)
3	Drive axis_1	SERVO		Switching on inhibited - set "OC/OFF2" = "1" (p0844, p0845)(42)

Slika 6.4 Dijagnostika uređaja

Web server omogućuje prikaz alarma te šalje povratnu informaciju korisniku. Svaki alarm ima zasebni kod pomoću kojeg se isti može identificirati. Sučelje prikazano na slici 6.5 sastoji se od 2 podsučelja: "*Diagbuffer*" i "*Alarm display*". Dijagnostički buffer služi za bilježenje svih događaja koji su se dogodili u nekom vremenskom periodu tijekom korištenja pretvarača poput preuzimanja ili slanja podataka, vremenski okvir trajanja događaja, moguće greške prilikom izvođenja programske logike itd. To uključuje događaje poput preuzimanja ili slanja podataka, trajanje tih događaja, izvođenje programske logike, moguće greške i slično. Dijagnostički buffer pruža povijest događaja koji se mogu koristiti za dijagnostiku, otkrivanje problema i praćenje performansi.

Prikaz alarma prikazuje sva alarmna stanja web servera koja se pojavljuju i sprema ih u memoriju upravljačke jedinice. Svaki alarm ima zasebni kod koji omogućuje identifikaciju i razumijevanje vrste problema koji se javlja. Alarmna stanja obično uključuju informacije o kritičnim ili neispravnim uvjetima rada pretvarača.

Time	Type	Drive object	Alarm
10.01.2000 23:50:31.689	A	Drive control	1073: POWER ON required for backup copy on memory card (0)
10.01.2000 23:27:31.487	A	Infeed_1	6400: Infeed: Line supply data identification selected/active (0)

Slika 6.5 Prikaz alarma i poruka od strane servera

Jedna od opcija koju nudi S120 web server jest kreiranje posebnih lista parametara koje će se izvršiti prilikom poziva. Svaka lista parametara mora sadržavati naziv liste, komponentu pretvarača čiji se parametri pozivaju te sami parametri komponente pretvarača. Liste je moguće spremati lokalno na računalo ili unutar memorije web servera. Bitovi parametara moraju odgovarati onima u konfiguraciji samog frekvencijskog pretvarača.

**Parameter - Define**

Define

**New parameter list**

Add new list

---

**Change/delete list**

List name  Position

DO  Parameter

Slika 6.6 Prikaz parametara web servera

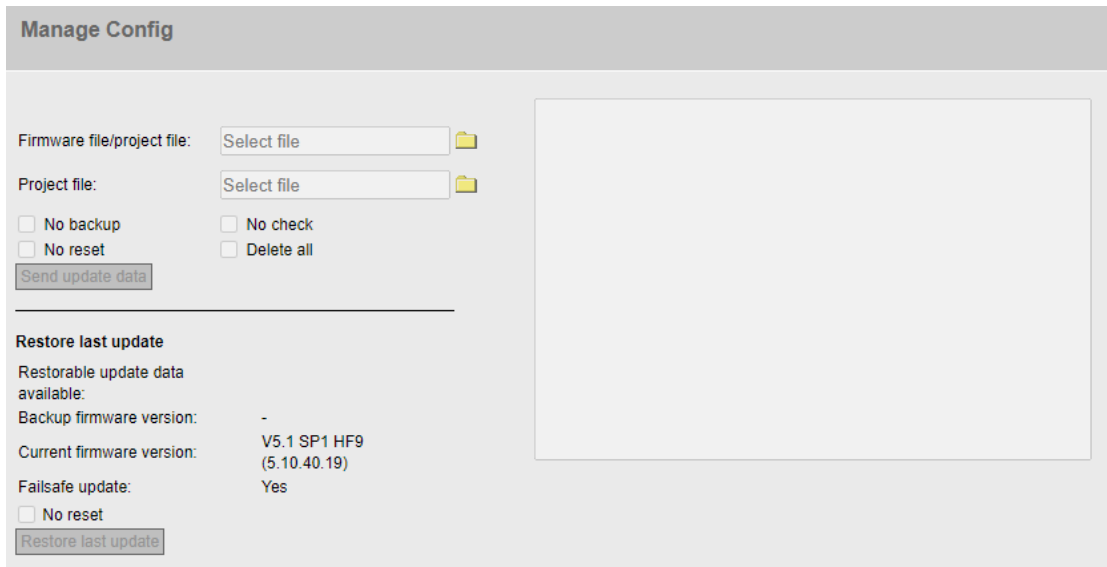
Naziv Liste Parametara označava jedinstveno ime za listu parametara u cilju jednostavnije identifikacije i upravljanja različitim listama koje možete stvoriti.

Komponenta Pretvarača odnosi se na određenu funkcionalnost ili dijelove frekvencijskog pretvarača na koje se odnose parametri na listi (upravljanje brzinom, kontrolu položaja, dijagnostiku).



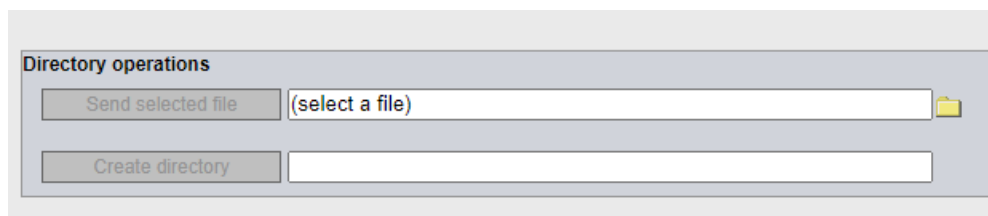
Parametri komponente pretvarača mogu biti postavke kao što su brzina, položaj, frekvencija, ili drugi parametri specifični za odabrano upravljanje.

Spremanje Lokalno ili Unutar Memorije Web Servera kao opcija omogućuje korisnicima da odaberu gdje će se spremati definirane liste parametara ovisno o potrebama i pristupu korisnika.



Slika 6.7 Postavljanje konfiguracije web servera

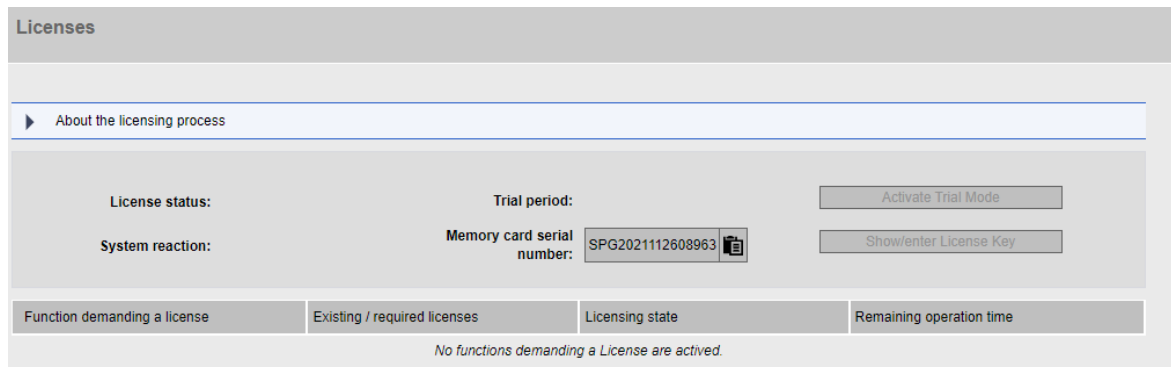
Konfiguraciju web servera moguće je obaviti automatski prilikom puštanja pretvarača u pogon ili na način da se prenese već unaprijed postavljena konfiguracija pretvarača. Svi preneseni projekti moraju biti u '.zip' formatu, u suprotnom web server neće moći očitati datoteke koje se prenose. Ukoliko se koristi projekt čija je firmware verzija starija od firmware pretvarača moguće je prenijeti i firmware datoteku zajedno sa projektom. Prenesene datoteke skladište se unutar lokalne memorije samog web servera te server nudi opciju kreiranja "Backup" datoteke.



Slika 6.8 Prenosjenje datoteka na server

Ukoliko postoji zahtjev za prijenos specifičnih datoteka koje nisu unutar sklopa projekta moguće je prenijeti ih na server koristeći "Directory Operations" sučelje. Ova opcija je iznimno bitna za kreiranje korisnički definiranih web sučelja koja će biti detaljnije razrađena u sljedećem poglavlju.

Sve licence uređaja prikazane su na slici 6.9. Serijski broj memorijske kartice koja se koristi služi za aktivaciju softvera te veže memorijsku karticu sa licencom i omogućuje korištenje softvera.



Slika 6.9 Prikaz licence uređaja

## 6.2. Korisnički definirane web stranice

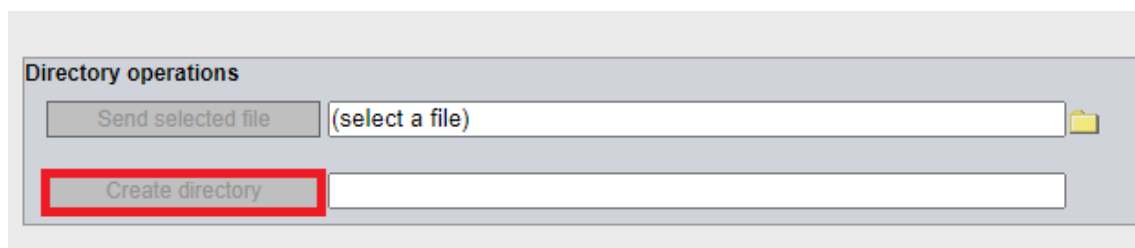
Jedna od glavnih značajki S120 web servera jest mogućnost kreiranja korisnički definiranih web stranica. Ove stranice mogu se koristiti za dijagnostiku, kontrolu, upravljanje i parametrisiranje upravljačkog sustava. Svaka stranica realizirana je putem HTML (engl. "*Hyper Text Markup Language*") koda, estetski dizajnirana koristeći CSS (engl. "*Cascading Style Sheet*") te funkcionalno uspostavljena putem Javascript-a. U uvodu poglavlja objašnjeno je kako S120 web server koristi MWSL programski jezik za očitavanje i implementaciju parametara pretvarača. MWSL kod se direktno implementira unutar koda stranice koristeći oznaku `<MWSL>`. Svaka stranica mora sadržavati sljedeće;

1. Izvor parametara – svi parametri pretvarača koji će biti prikazani na stranici moraju imati jasno definiran izvor fiksne strukture. Parametri se čitaju sa pretvarača pozivom MWSL funkcija te svaki parametar ima alociran zasebni identifikacijski faktor (ID).
2. Glavna stranica – na ovoj stranici nalazi se sav sadržaj koji se prikazuje. Prethodno definirani parametri pretvarača mogu biti pozvani u sadržaj glavne stranice putem odgovarajućeg identifikacijskog faktora. Bitno je naglasiti kako identifikacijski faktor HTML elementa mora odgovarati odgovarajućem identifikacijskom elementu sa izvora parametara. Putem Javascript-a svi podaci se ciklički ažuriraju te dinamički prikazuju na glavnoj stranici web servera.

3. Stranica za učitavanje – omogućuje korištenje skripti za dinamičku realizaciju glavne stranice. Glavna uloga ove stranice jest integracija Javascript funkcija unutar glavne stranice što omogućuje prethodno spomenuto cikličko ažuriranje podataka.
4. Javascript datoteka – sve funkcije koje se koriste prilikom izrade web stranice nalaze se unutar ove datoteke. Unutar datoteke nalaze se predefimirane i korisnički definirane funkcije koje se izvršavaju na glavnoj stranici nakon poziva.
5. CSS datoteka – služi za estetsko modeliranje stranice, odnosno omogućuje formatiranje izgleda same web stranice. Web stranica ne mora nužno sadržavati ovu datoteku, ali integracija CSS-a prilikom dizajna web stranice znatno olakšava dizajniranje estetski prikladnog sučelja.

Za početak dizajniranja web stranice potrebno je kreirati sljedeće datoteke na web serveru;

- CSS
- Configuration
- Images
- Scripts



Slika 6.10 Stvaranje direktorije

Navedene datoteke nalazit će se unutar glavne datoteke nazvane "WEBSITES". Web server na ovaj način ima jasno definiranu lokaciju direktorije odakle može učitavati kreirane stranice. Unutar ovih datoteka nalazit će se sve potrebne skripte, funkcije i slike koje će biti prikazane, odnosno pozvane na glavnoj stranici. Odabirom opcije "Send selected file" moguće je prenijeti već unaprijed kreiranu datoteku na web server.

Slika 6.11 predstavlja programski kod kreiranja izvora parametara. Ova struktura je fiksna i ne može se mijenjati osim u slučajevima gdje korisnik ne koristi određen tip parametara. U tom slučaju izvor tog parametra mora biti obrisan.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
  <versions>
    <document>1</document>
    <minInterpreter>1</minInterpreter>
  </versions>
  <variables>
    <user>
      <MWSL><!--
      var PREFIX_DO_PARAMS = "",
      doName = GetVar("doName", "URL");
      PREFIX_DO_PARAMS = doName + ".Params.";
      --></MWSL>
    </user>
    <plainTextVariables>
      <!-- if no plainTextVariables are defined,
      this section has to be deleted-->
    </plainTextVariables>
    <numericVariables>
      <!-- if no numericVariables are defined,
      this section has to be deleted-->
    </numericVariables>
    <indicatorVariables>
      <!-- if no indicatorVariables are defined,
      this section has to be deleted-->
    </indicatorVariables>
    <gaugeVariables>
      <!-- if no gaugeVariables are defined,
      this section has to be deleted-->
    </gaugeVariables>
    <barVariables>
      <!-- if no barVariables are defined,
      this section has to be deleted-->
    </barVariables>
  </variables>
</root>
```

Slika 6.11 Kreiranje izvora parametara [12]

Koristeći MWSL definira se varijabla koja će biti prikazana na glavnoj stranici. Varijabla je pozvana na glavnu stranicu koristeći ID MWSL varijable.

Main page	Parameter source
<pre>&lt;div&gt;   &lt;table&gt;     &lt;tr&gt;       &lt;td&gt;Actual Speed&lt;/td&gt;       &lt;td id="valActualSpeed"&gt;---&lt;/td&gt;       &lt;td&gt;1/min&lt;/td&gt;     &lt;/tr&gt;   &lt;/table&gt; &lt;/div&gt;</pre>	<pre>&lt;plainTextVariables&gt; &lt;valActualSpeed&gt;   &lt;MWSL&gt;&lt;!--   ...   --&gt;&lt;/MWSL&gt; &lt;/valActualSpeed&gt; &lt;/plainTextVariables&gt;</pre>

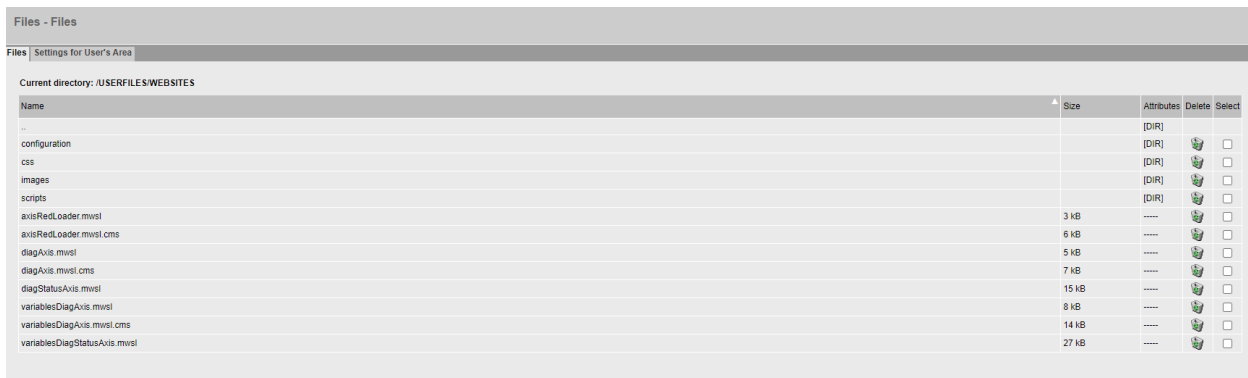
Slika 6.12 Primjer povezivanja izvora parametara sa HTML kodom glavne stranice [12]

Na slici 6.13 prikazano je paralelno grafičko sučelje web stranice i HTML programsku realizaciju samog sučelja. Kreirana je klasa *"indicatorItem"* unutar koje spadaju sve prikazane varijable. Elementi klase poprimaju sve vrijednosti navedene za tu klasu. Sintaksa `<div>` označava odjeljke unutar HTML koda (može se koristiti analogija skladišta) gdje svaki odjeljak poprima klasu i svaka klasa poprima odgovarajući MWSL identifikacijski faktor. Također su povezane slike indikatora koje bojom označavaju stanje varijable. Sintaksa `<p>` označava tekstni odlomak gdje se ispisuje tekst na glavnoj stranici.

HTML code	Web page
<pre> &lt;div class="pageColumn" id="indicatorsColumn"&gt;   &lt;h1&gt;Enables&lt;/h1&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorEnablesAvailable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;Enables existing&lt;/p&gt;   &lt;/div&gt;   &lt;br /&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorOff1Enable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;OFF1 enable&lt;/p&gt;   &lt;/div&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorOff2Enable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;OFF2 enable&lt;/p&gt;   &lt;/div&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorOff3Enable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;OFF3 enable&lt;/p&gt;   &lt;/div&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorPulseEnable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;Operation enabled&lt;/p&gt;   &lt;/div&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorRampFctGeneratorEnable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;Ramp-function generator enable&lt;/p&gt;   &lt;/div&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorStartStopRampFctGenerator"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;Continue ramp-function generator&lt;/p&gt;   &lt;/div&gt;   &lt;div class="indicatorItem"&gt;     &lt;img id="indicatorSetpointEnable"       src="images/indicatorOff.png" alt="off" /&gt;     &lt;p&gt;Speed setpoint enable&lt;/p&gt;   &lt;/div&gt; &lt;/div&gt; </pre>	

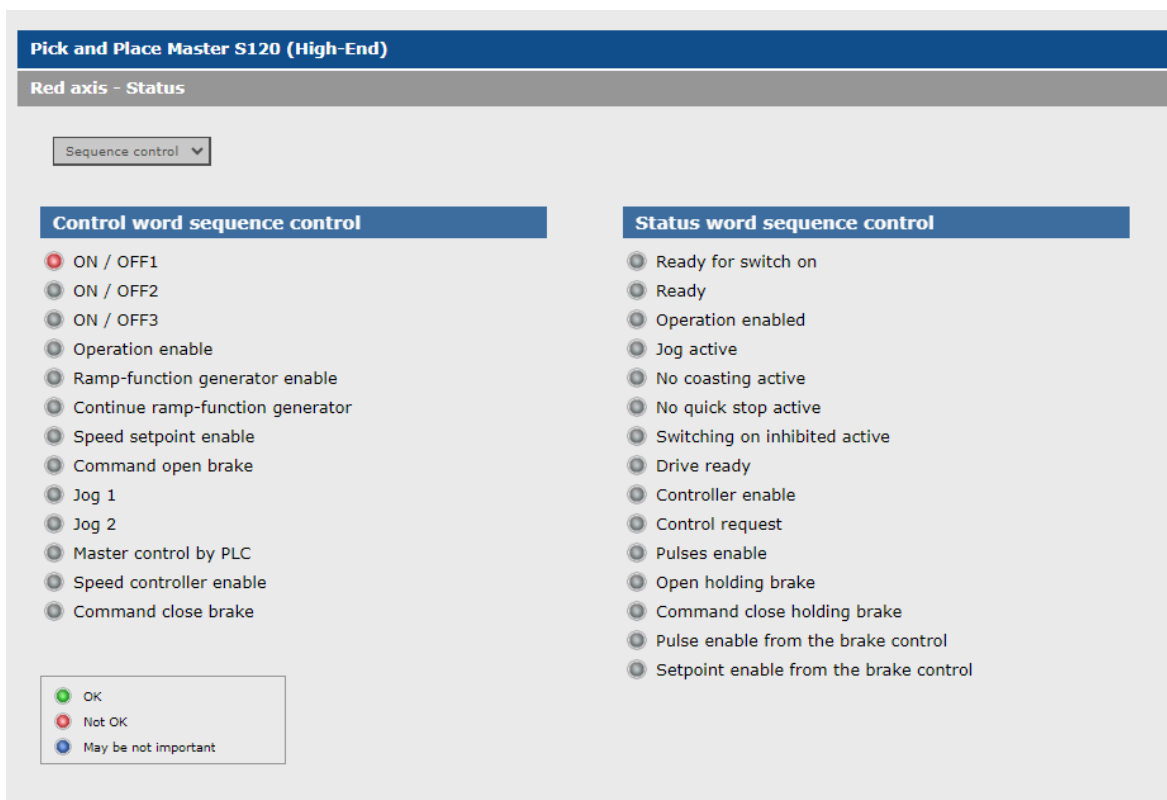
Slika 6.13 Paralelni prikaz HTML koda i grafičke realizacije na stranici [12]

Sve datoteke moraju biti prenesene u odgovarajuće direktorije. Redoslijed datoteka je također bitan. Moguće je koristiti bilo koje programsko razvojno okruženje za pristup kodu. Na slici 6.14 prenesene su datoteke za konfiguraciju dijagnostičkog sučelja parametara motora razvijenog od strane Siemens.



Slika 6.14 Sve korištene datoteke za izradu web stranice

Prilikom prvog postavljanja programskog koda bez unaprijed definirane konekcije između parametara pretvarača sa web sučeljem rezultira izgledom web sučelja prikazanog na slici 6.15. Varijable potrebno naknadno povezati sa identifikacijskim faktorom pretvarača.

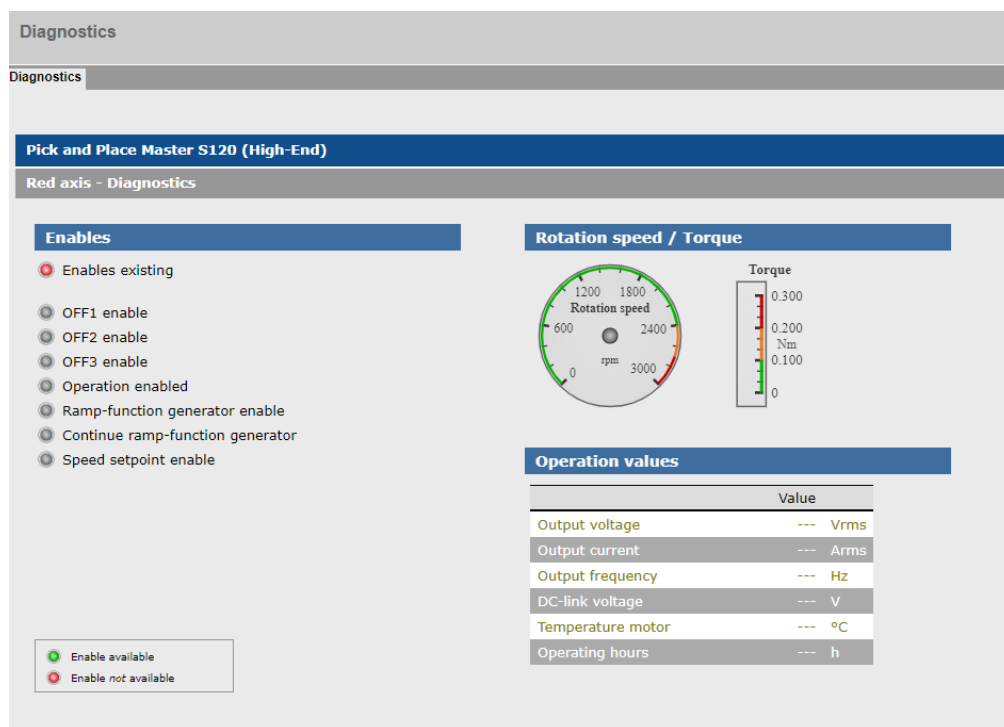


Slika 6.15 Izgled predefiniiranog web sučelja

U korisničkom sučelju "*Diagnostics*" moguće je vizualno prikazati brzinu motora i moment motora. U dijagnostičkom prikazu u ovom radu (slika 6.16) prikazane su sljedeće varijable:

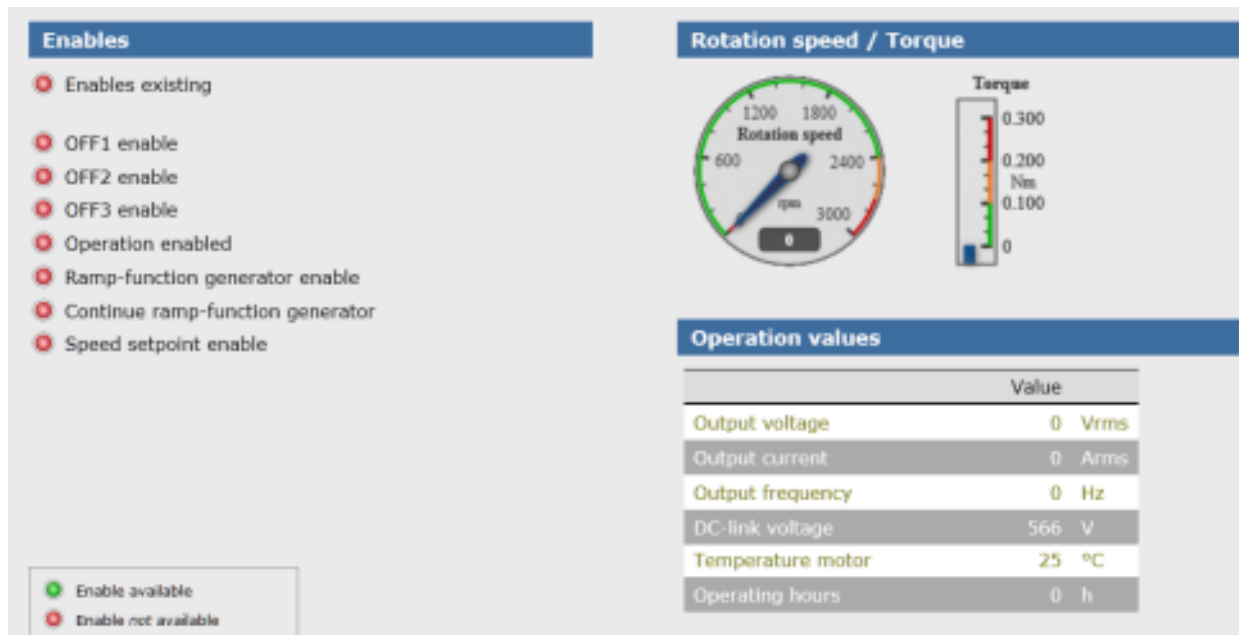
1. Izlazni napon [V – rms]
2. Izlazna struja [A - rms]
3. Frekvencija [Hz]
4. Napon istosmjernog međukruga [V]
5. Temperatura motora [C]
6. Vrijeme upravljanja sučeljem [h]

Posebno je definirana varijabla "*Enables*" koja nakon dobivanja signala za aktivaciju pokreće sve ostale prikazane varijable. Kao i u prethodnom slučaju parametri pretvarača moraju biti povezani sa odgovarajućim identifikacijskim faktorom unutar programskog koda. Moguće je izmjenjivati koja se web stranica prikazuje odabirom na "*Diagnostics*" ili na "*Status*".



Slika 6.16 Dijagnostički prikaz

Nakon povezivanja parametara pretvarača sa web serverom na slici 6.17 prikazano je kako su sve varijable označene kao nedostupne. Potrebno je povezati sve prikazane varijable sa odgovarajućim ulaznim bitovima parametra pretvarača označenim "rx.y".



Slika 6.17 Konfiguracija varijabli dijagnostičkog prikaza

Naziv varijable	Opis varijable
SERVO_03.r35	Temperatura motora
SERVO_03.r63	Brzina vrtnje motora
SERVO_03.r65	Pozicija motora
SERVO_03.r72	Moment motora
SERVO_03.r77	Akceleracija motora

Tablica 1. Korištene varijable i njihovi nazivi

Konfiguracija grafičkog prikaza brzine vrtnje motora i momenta motora kako je prikazano na slici 6.17 u programskom kodu označeno je sa identifikacijskim faktorima "gaugeSpeed" unutar klase "canvasControl" te "valOutputVoltage".

Identifikacijski faktori predstavljaju oznake ili imena koja se koriste za referenciranje određenih grafičkih elemenata unutar programskog koda.

Klasa "canvasControl" predstavlja vrstu kontrolnog elementa ili objekta unutar programskog koda koja može sadržavati metode i funkcije za upravljanje i konfiguriranje grafičkog prikaza.



Povezivanje identifikacijskih faktora unutar koda web stranice vrši se na sljedeći način.

- 1) Definira se odjeljak klase *"pageColumnRight"* sa MWSL identifikacijskim faktorom *"valuesColumn"*. Odjeljak služi za centriranje prikaza parametara pretvarača na način kako je prikazano na slici 6.18.
- 2) Unutar odjeljka definira se odjeljak *"Rotation speed / Torque"* koji prikazuje naziv samog odjeljka.
- 3) Kreira se dodatni `<div>` element unutar odjeljka sa identifikacijskim faktorom *"canvasControls"* koji objedinjuje sve klase i identifikacijske faktore koji se pozivaju unutar odjeljka, zajedno sa parametrima koji služe za dizajniranje izgleda samog odjeljka.
- 4) Pošto se radi o grafičkom prikazu koristi se `<canvas>` parametar čija je klasa definirana kao *"canvasControl"* i identifikacijski faktor je MWSL parametar *"gaugeSpeed"*. Pozivom ovog parametra prikazuje se grafičko sučelje brzine vrtnje motora. Visina i širina postavljeni su na *140px* sa *140px*.

Na isti način konfiguriran je MWSL parametar *"barTorque"* za prikaz aktualnog momenta motora. Ovaj način deklaracije, stila i poziva parametara mora biti poštovan za sve aktualne parametre pretvarača. U primjeru prikazanom na slici 6.17 motor nije pokrenut te sukladno tome brzina vrtnje nije prikazana.

```
<div class="pageColumnRight" id="valuesColumn">
  <h1>Rotation speed / Torque</h1>
  <div id="canvasControls">
    <canvas class="canvasControl" id="gaugeSpeed" width="140"
      height="140">
    </canvas>
    <span> </span>
    <canvas class="canvasControl"
      id="barTorque"
      width="80"
      height="140"></canvas>
  </div>
</div>
```

Slika 6.18 Programski kod uspostavljanja grafičkog prikaza brzine vrtnje motora

Unutar istog `<div>` odjeljka prikazanog na slici 6.18 kreiran je dodatni odjeljak *"Operation values"* koji prikazuje sve parametre pretvarača. Kreirana je tablica klase *"valuesTable"* unutar koje se kreira `<colgroup>` tip koji služi za deklaraciju više podgrupa istog stupca unutar tablice.

Klase ovih grupa označene su sa *"valuesTableColumn1"*, *"valuesTableColumnValues"* i *"valuesTableColumnUnits"*. Identifikacijski parametar MWSL koda koji se poziva je *"valOutputVoltage"* za prikaz napona motora. Ista analogija vrijedi za sve naknadne parametre.

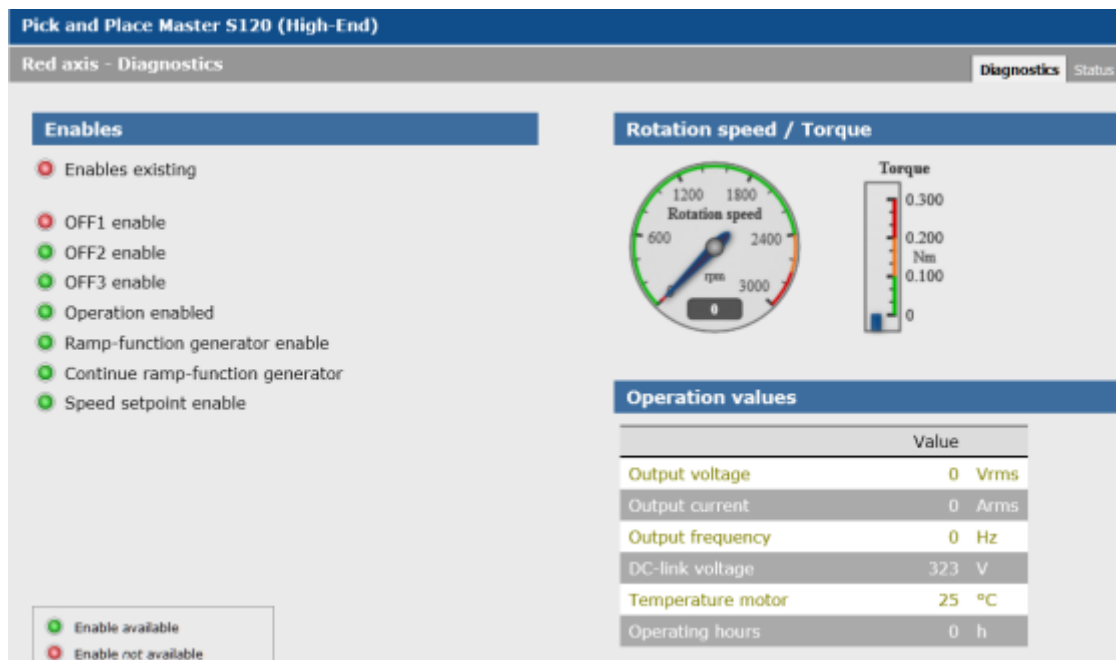
```

<div class="pageColumnRight" id="valuesColumn"
>
  <h1>Operation values</h1>
  <table class="valuesTable">
    <colgroup>
      <col class="valuesTableColumn1" />
      <col class
        ="valuesTableColumnValues" />
      <col class
        ="valuesTableColumnUnits" />
    </colgroup>
    <thead>
      <tr>
        <th></th>
        <th>Value</th>
        <th></th>
      </tr>
    </thead>
    <tbody>
      <tr class="valuesTableRowEven">
        <td>Output voltage</td>
        <td id="valOutputVoltage"
          class="valuesTableValue">
          <span class="value"
            >----</span>
          <span class="warningLimit"
            >300</span>
          <span class
            ="criticalLimit">320
          </span>
        </td>
        <td>Vrms</td>
      </tr>
    </tbody>
  </table>
</div>

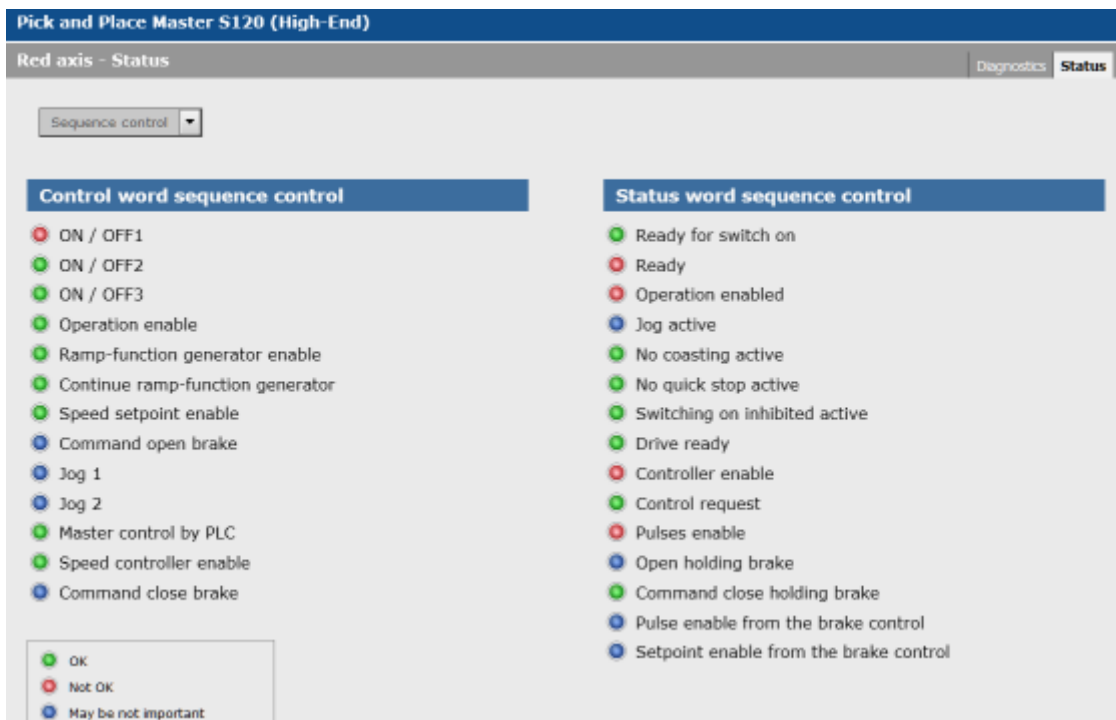
```

Slika 6.19 Programski kod uspostavljanja prikaza parametara pretvarača

Ispravno uspostavljanje web stranice S120 pretvarača, na osnovu prethodno obrazloženog postupka za povezivanje identifikacijskih faktora unutar koda web stranice, prikazano je na slikama 6.20 i 6.21. Pokretanje motora rezultira aktivacijom statusnih signala i promjene grafičkog prikaza motora.



Slika 6.20 Dijagnostička stranica [12]



Slika 6.21 Status stranica [12]

## 7. ZAKLJUČAK

U ovom radu je prikazan proces dizajniranja, konfiguracije i implementiranja interaktivnog korisničkog sučelja frekvencijskog pretvarača za analitički i grafički prikaz parametara i performansi izmjeničnog motora. Frekvencijski pretvarači iznimno su svestrani uređaji koji omogućuju pristup i upravljanje raznim parametrima motora. Za potrebe upravljanja pogona sinkronog motora s permanentnim magnetima serije 1FK7 sa ugrađenim enkoderom koristio se frekvencijski pretvarač Siemens Sinamics S120 koji u sebi ima ugrađen firmware za podešavanje upravljačkog web servera i PLC procesorske jedinice.

Konfiguracija parametara pretvarača obavljena je koristeći Siemensov software alat STARTER unutar okruženja TIA Portala koji omogućuje jednostavan pristup konfiguraciji sučelja pretvarača. Korištenje opcije automatske detekcije software pronalazi sve spojene komponente pretvarača u predmemoriji podataka te ih dodaje u integrirano sučelje. U slijedećem koraku je moguća dodatna konfiguracija pretvarača te je slijedilo spremanje i prenošenje same konfiguracije na PLC. U ovom radu se parametrirao pogon jednog motora no dizajn frekvencijskih pretvarača omogućuje i upravljanje pogona od više motora istovremeno.

Prije puštanja u pogon uspostavljena fizička i programska PROFINET veza između frekvencijskog pretvarača i PLC uređaja te su postavljene odgovarajuće IP adrese putem PG/PC sučelja. Programska logika upravljanja motorom realizirana je putem PLC-a te je kreirano HMI sučelje koje vizualno prikazuje upravljanje motorom. Nakon testiranja konfiguracije parametara pretvarača i kreiranja programske logike upravljanja motorom uspostavljena je veza sa web serverom pretvarača koristeći predefiniranu IP adresu priključka na koji je pretvarač spojen sa računalom. Jedna od glavnih karakteristika spomenutog web servera jest mogućnost kreiranja korisnički definiranih web stranica koje služe za dijagnostiku i status upravljanog uređaja, kao i za prikaz mogućih alarma prilikom kvara. Za ispravno povezivanje parametara pretvarača sa web serverom potrebna je implementacija MWSL koda.

Zaključak rada je da programsko sučelje koje u pravilu dolazi kao sastavni dio podrške frekvencijskih pretvarača osigurava korisniku pristup kreiranju topologije upravljanog pogona izmjeničnih motora korištenjem web sučelja. Vizualizacija parametara i veličina potrebnih za upravljanje omogućeno je kroz široki spektar grafičkih vizualizacija i prekidača za provedbu daljinskog nadzora i upravljanja električnog pogona što je od iznimne važnosti u suvremenim automatiziranim industrijskim pogonima.

## LITERATURA

- [1] Danfoss, Najvažnije o frekvencijskim pretvaračima, Graphis, Zagreb, 2013.
- [2] Danfoss handbook: Facts Worth Knowing about AC Drives, 2016.  
<https://www.danfoss.com/> (pristup: 13.09.2023.)
- [3] Sinkroni i asinkroni električni strojevi, I. Mandić, V. Tomljenović, M. Pužar, Zagreb, 2012.
- [4] <https://press.siemens.com/global/en/pressrelease/sinamics-s120-drive-system-firmware-and-hardware-updates-increase-user-friendliness> (pristup: 13.09.2023.)
- [5] SINAMICS S120 Manual Control Units and additional system components;  
<https://support.industry.siemens.com/cs/document/109478725> (pristup: 13.09.2023.)
- [6] [https://upb-lea.github.io/gym-electric-motor/parts/physical\\_systems/converters/B6C.html](https://upb-lea.github.io/gym-electric-motor/parts/physical_systems/converters/B6C.html)  
(pristup: 11.11.2023.)
- [7] <https://press.siemens.com/global/en/pressrelease/sinamics-s120-drive-system-firmware-and-hardware-updates-increase-user-friendliness>(pristup: 13.09.2023.)
- [8] 1FK7 Synchronous Motors SINAMICS S120 Configuration Manual, 2006
- [9] Programabilni logički kontroleri, TVZ, G. Malčić
- [10] <https://www.automation24.biz/siemens-simatic-dp-cpu-1510sp-f-1-pn-6es7510-1sj01-0ab0> (pristup: 13.09.2023.)
- [11] Maršić Danijel, Goran Malčić, Ivica Vlašić: “Izrada programskih komponenti u TIA Portal programskom okruženju.” MIPRO, 2014.
- [12] SINAMICS S120 web server - Creating user-defined web pages,  
<https://support.industry.siemens.com/cs/document/68691599> (pristup: 13.09.2023.)

## SAŽETAK

U diplomskom radu objašnjen je princip funkcioniranja frekvencijskog pretvarača S120 i PLC uređaja za puštanje u pogon sinkronog motora s permanentnim magnetima. Programska logika upravljanja motorom razrađena je u PLC softveru TIA Portal dok je konfiguracija i parametrizacija pretvarača obavljena koristeći softverski alat STARTER. Frekvencijski pretvarač ima integrirani web server koji omogućuje kreiranje korisnički definiranih web stranica za analitički i grafički prikaz parametara motora. Na kraju je izrađena web stranica za prikaz parametara motora te je objašnjen princip implementacije parametara pretvarača na web server.

**Ključne riječi:** frekvencijski pretvarač, S120, TIA Portal, STARTER, korisnički definirane web stranice, MWSL, PLC

## **ABSTRACT**

### **Configuration and control of a Siemens Sinamics S120 frequency converter via the web server.**

In the master's thesis the operating mode and commissioning of the S120 frequency converter and PLC is explained in order to commission the synchronous motor with permanent magnets. The program logic to control the motor was developed in the PLC software TIA Portal while the configuration and parametrization of the frequency converter was done using the software tool STARTER. The frequency converter has an in-built web server which allows the user to create user-defined web pages for analytical and graphical representation of the motor parameters. In the end of the master's thesis a user-defined web page was created to display the motor parameters and the implementation of said parameters to the web server was explained.

**Key words:** frequency converter, S120, TIA Portal, STARTER, user-defined web pages, MWSL, PLC

## **ŽIVOTOPIS**

Fran Dujmović rođen je 14. prosinca 1999. godine u Osijeku gdje je pohađao osnovnu školu Ivana Filipovića te 2014. godine upisuje II gimnaziju u Osijeku. Nakon završenog srednjoškolskog obrazovanja 2018. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, gdje 2021. godine završava preddiplomski sveučilišni studij i upisuje diplomski smjer Industrijska elektroenergetika iste godine.

---

Potpis autora