Industrijsko vizualizacijsko sučelje za nadzor i upravljanje robotskim sustavom

Lozančić, Ivan

Master's thesis / Diplomski rad

2023

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:898289

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Ivan Lozančić

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Marko Švaco, mag. ing. mech.

Student:

Ivan Lozančić

Zagreb, 2023.

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

Zahvaljujem se svim profesorima na znanju koje su mi prenijeli tijekom perioda studiranja. Posebice bi se želio zahvaliti mentoru doc. dr. sc. Marku Švaci na pruženoj podršci i usmjeravanju tijekom izrade diplomskog rada. Također bi se želio zahvaliti kolegama iz tvrtke Strojotehnika koji su znatno pridonijeli svojim znanjem i savjetima. Za kraj bih se, naravno, želio zahvaliti svojoj obitelji na svoj pruženoj podršci kroz sve ove godine moga obrazovanja. Bez njihove podrške, ne bih uspio postići ovaj važan cilj u svom akademskom putu.

Ivan Lozanči

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602 – 04 / 23 – 6 / 1	
Ur. broj: 15 - 1703 - 23 -	

DIPLOMSKI ZADATAK

Student: **IVAN LOZANČIĆ** Mat. br.: 0035205699

Naslov rada na hrvatskom jeziku: **Industrijsko vizualizacijsko sučelje za nadzor i upravljanje robotskim sustavom**

Naslov rada na engleskom jeziku: **Industrial visualization interface for monitoring and controlling of a robotic system**

Opis zadatka:

Korisnička sučelja za nadzor i upravljanje robotiziranim stanicama važan su čimbenik učinkovite automatizacije. Iako robotizirane stanice rade bez ljudi postoje razni aspekti kada je ljudska intervencija neophodna. To su slučajevi puštanja u rad, pregleda grešaka i trenutnog stanja rada stanice, intervencija uslijed neplaniranih zastoja i sl. Kako bi se navedene radnje mogle izvršiti učinkovito i pravovremeno nužna su napredna industrijska korisnička i vizualizacijska sučelja za nadzor i upravljanje robotskim sustavima.

U sklopu ovog rada potrebno je:
- Omogućiti upravljanje i nadzor industrijskim i kolaborativnim robotima i transportnim sustavima u Laboratoriju za autonomne sustave u CRTA-i pomoću Siemens „SIMATIC WinCC Unified" sustava,
- Roboti trebaju biti podijeljeni u dva modula (svaki modul organiziran oko jednog transportnog sustava), svaki modul treba biti upravljan sa zasebnim PLC-om, dok je komunikacija između PLC-a putem Modbus TCP/IP protokola,
- Potrebno je napraviti vizualizacijski sustav i sučelje koje će omogućiti nadzor rada cijelog sustava s udaljenog mjesta i putem bežične mreže, kao i upravljanje i nadzor svakog robota zasebno,
- Omogućiti jednostavnu nadogradnju vizualizacijskog sučelja,
- Izraditi biblioteku funkcijskih blokova za Siemens PLC-e u Laboratoriju kako bi se svaki robot mogao modularno definirati u sklopu vizualizacijskog sučelja.

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

Zadatak zadan: 17. studenog 2022.

Rok predaje rada: 19. siječnja 2023.

Predviđeni datum obrane: 23. siječnja do 27. siječnja 2023.

Zadatak zadao: doc. dr. sc. Marko Švaco

Predsjednica Povjerenstva: prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ I
POPIS SLIKA II
SAŽETAK IV
SUMMARY V
1. UVOD 1
1.1. Proizvodni sustavi u industriji 2
2. OPIS KOMPONENTI ROBOTSKOG SUSTAVA 4
2.1. PLC i popratni moduli 7
2.2. Roboti 9
2.3. Komunikacija između komponenti 12
3. IZRADA I MODIFIKACIJA PROGRAMSKE PODRŠKE SUSTAVA 16
3.1. Programska podrška PLC-a 16
3.1.1. Glavni organizacijski blok 17
3.2. Programska podrška korisničkog sučelja 23
3.3. Programska podrška robota 35
3.3.1. Fanuc kolaborativni roboti 35
3.3.2. UR kolaborativni robot 47
3.3.3. ABB kolaborativni robot 54
3.3.4. KUKA industrijski robot – Agilus KR10 R1100-2 59
4. TESTIRANJE I PUŠTANJE U POGON 66
5. MODULARNOST SUSTAVA 68
6. ZAKLJUČAK 69

POPIS SLIKA

Slika 1. Mogućnost pristupa WinCC Unified korisničkom sučelju preko bilo kojeg uređaja s web preglednikom [1] ... 2
Slika 2. Primjer automatizirane proizvodne linije u industriji [2] ... 2
Slika 3. Rexroth modul s označenim proizvodnim stanicama ... 4
Slika 4. Montrac modul s označenim proizvodnim stanicama ... 5
Slika 5. Blokovski dijagram proizvodnog sustava s označenim proizvodnim stanicama ... 6
Slika 6. Topološki prikaz cijelog sustava unutar TIA Portal projekta ... 7
Slika 7. Prikaz PLC-a 1215C DC/DC/DC i SCALANCE XB005 *switch*-a na Montrac modulu ... 8
Slika 8. Modularni sustav SIMATIC ET200SP s po dvije ulazne i izlazne kartice na Montrac modulu ... 8
Slika 9. Elektronički ventilski blok EX260 SPN 1/2 na Montrac modulu ... 9
Slika 10. ABB YuMi robot ... 10
Slika 11. UR3 robot ... 10
Slika 12. Fanuc CR-15iA robot ... 11
Slika 13. Fanuc CRX-10iA robot ... 11
Slika 14. UR5 robot ... 12
Slika 15. Prikaz definirane podmreže u TIA Portal projektu ... 13
Slika 16. Prikaz Montrac PROFINET IO sustava u TIA Portal projektu ... 14
Slika 17. Prikaz definiranih HMI veza između WinCC Unified PC-a i PLC-a ... 14
Slika 18. Naglašeni prikaz HMI veze između WinCC Unified PC-a i Montrac PLC-a u TIA Portal projektu ... 15
Slika 19. Struktura Main OB-a Rexroth PLC-a u TIA Portal projektu ... 17
Slika 20. Mreža za upravljanje stanicom 1 na Rexroth modulu ... 19
Slika 21. Mreža za HMI upravljanje robotom ovisno o načinu rada sustava unutar funkcijskog bloka stanice ... 20
Slika 22. Funkcija za kontrolu transportnim sustavom s dodatnim ulaznim signalima unutar funkcijskog bloka stanice ... 21
Slika 23. Logika za odabir načina rada ovisno o sigurnosti sustava unutar funkcijskog bloka „SigurnostSustava“ ... 22
Slika 24. Sučelje „WinCC Unified Configuration“ aplikacije ... 23
Slika 25. Postavke korisničke administracije „WinCC Unified Configuration“ aplikacije ... 24
Slika 26. Primjena postavki „WinCC Unified Configuration“ aplikacije ... 25
Slika 27. Sučelje „WinCC Unified Certificate Manager“ aplikacije ... 26
Slika 28. Web stranica korisničkog sučelja nakon konfiguracije ... 27
Slika 29. Početni zaslon korisničkog sučelja ... 28
Slika 30. Opis funkcija tipkala na zaslonu Rexroth modula ... 28
Slika 31. Zaslon s dnevnikom alarma ... 29
Slika 32. Skočni zaslon s aktivnim alarmima ... 29
Slika 33. Definiranje alarmne tekstualne liste ... 30
Slika 34. Definiranje okidača alarma ... 30
Slika 35. Zaslon s postavkama rada sustava ... 31
Slika 36. Standardizirani zaslon za robote ... 32
Slika 37. Logika za definiranje prekidača s prikazom stanja ... 33
Slika 38. Skočni prozor za upozorenje pri pomicanju robota u početni položaj ... 34
Slika 39. Simulacija CRX-10iA robota pomoću ROBOGUIDE programskog alata ... 35
Slika 40. Konfiguracija digitalnih izlaza Fanuc robota za komunikaciju s PLC-om ... 36
Slika 41. Konfiguracija digitalnih izlaza i ulaza Fanuc robota na strani PLC-a ... 36

Slika 42. UOP ulazni signali za kontrolu Fanuc robota preko PLC-a 37
Slika 43. Konfiguracija postavki na „Prog Select“ zaslonu Fanuc robota 38
Slika 44. Definirani PNS0001 program Fanuc robota 38
Slika 45. Definiranje početnog položaja Fanuc robota 39
Slika 46. Spremanje početnog položaja Fanuc robota u registar položaja 39
Slika 47. Definiranje *homing* programa za Fanuc robote 40
Slika 48. Definiranje makro naredbe *homing* programa Fanuc robota 40
Slika 49. Definiranje COND_ERROR programa Fanuc robota 41
Slika 50. Definiranje ERROR_OUTPUT programa Fanuc robota 41
Slika 51. Funkcijski blok za upravljanje i nadzor Fanuc robota 42
Slika 52. Logika za ispisivanje statusa Fanuc robota na HMI-u 43
Slika 53. Logika za inicijalizaciju Fanuc robota 44
Slika 54. Logika za slanje Fanuc robota u početni položaj 44
Slika 55. Logika za provjeru stanja Fanuc robota 45
Slika 56. Logika za upravljanje Fanuc robotom u automatskom načinu rada sustava 46
Slika 57. Definiranje statusne trake za Fanuc CRX-10iA robota 47
Slika 58. Potpuno definirani zaslon Fanuc CRX-10iA robota 47
Slika 59. Konfiguracija sistemskih ulaznih i izlaznih signala na UR robotima 48
Slika 60. Konfiguracija varijabli za daljinsko upravljanje UR robota 48
Slika 61. Odabir glavnog programa i ulaznog signala za inicijalizaciju UR robota 49
Slika 62. Konfiguracija PROFINET-a na UR robotu 49
Slika 63. Definirani glavni program UR robota 50
Slika 64. Funkcijski blok UR robota 51
Slika 65. Sekvenca za inicijalizaciju UR robota 52
Slika 66. Odabir programskog slučaja na UR robotu 53
Slika 67. Potpuno definirani zaslon UR3 robota 54
Slika 68. Sučelje ABB-ovog programskog alata RobotStudio 54
Slika 69. Definiranje ulaznih i izlaznih signala za ABB YuMi robota 55
Slika 70. Pridjeljivanje akcija sistemskim signalima ABB YuMi robota 55
Slika 71. Definiranje početnog položaja ABB YuMi robota 56
Slika 72. Pridjeljivanje „Power On“ eventa „power_on“ rutinama u RobotStudio-u 56
Slika 73. Pozivanje *homing* rutine na početku glavne petlje ABB YuMi robota 57
Slika 74. Logika za inicijalizaciju ABB robota 58
Slika 75. Potpuno definirani zaslon ABB YuMi robota 59
Slika 76. Sučelje KUKA WorkVisual programskog alata 59
Slika 77. Mapiranje ulaznih i izlaznih signala KUKA Agilus robota 60
Slika 78. Definiranje signala za upravljanje programom KUKA robota u automatskom načinu rada 61
Slika 79. Definiranje *switch – case* logike unutar „cell“ programa KUKA robota 62
Slika 80. Definiranje „pause“ programa KUKA robota 62
Slika 81. Mapiranje signala za daljinsko upravljanje KUKA robotom [6] 63
Slika 82. Dijagram signala za automatsko daljinsko upravljanje KUKA robotom [6] 64
Slika 83. Sekvenca za inicijalizaciju KUKA robota 64
Slika 84. Logika za odabir programskog slučaja vraćanja KUKA robota u početni položaj 65
Slika 85. Potpuno definirani zaslon KUKA Agilus robota 65
Slika 86. Definirani zaslona za upravljanje transportnim sustavima 66
Slika 87. Prikaz sustava spremnog za automatski način rada 67

SAŽETAK

U današnje vrijeme, u svrhu povećanja produktivnosti, smanjenja troškova te povećanja kvalitete proizvoda, automatizacija je neizbježan proces u gotovo svim industrijama. Međutim, za uspješnu primjenu automatizacije kod složenih industrijskih sustava nužno je imati kvalitetno korisničko sučelje koje će omogućiti lakši prijenos informacija između sustava i operatera te samim time riješiti niz problema razvoja automatizacije. Potaknuti time, odlučeno je napraviti industrijsko korisničko sučelje za nadzor i upravljanje industrijskih i kolaborativnih robota te transportnih sustava u Laboratoriju za autonomne sustave u sklopu Regionalnog centra izvrsnosti za robotske tehnologije (CRTA). Cilj je diplomskog rada prikazati proces izrade takvog korisničkog sučelja pomoću Siemensovog „SIMATIC WinCC Unified“ sustava te prikazati prednosti korištenja istog. Osim izrade korisničkog sučelja, prikazana je izrada i odgovarajućeg PLC programa te funkcijskih blokova u svrhu standardizacije dijelova sustava radi lakšeg modularnog korištenja istih u budućnosti.

Ključne riječi: automatizacija, industrijsko korisničko sučelje, CRTA, SIMATIC WinCC Unified, PLC, standardizacija, modularno korištenje

SUMMARY

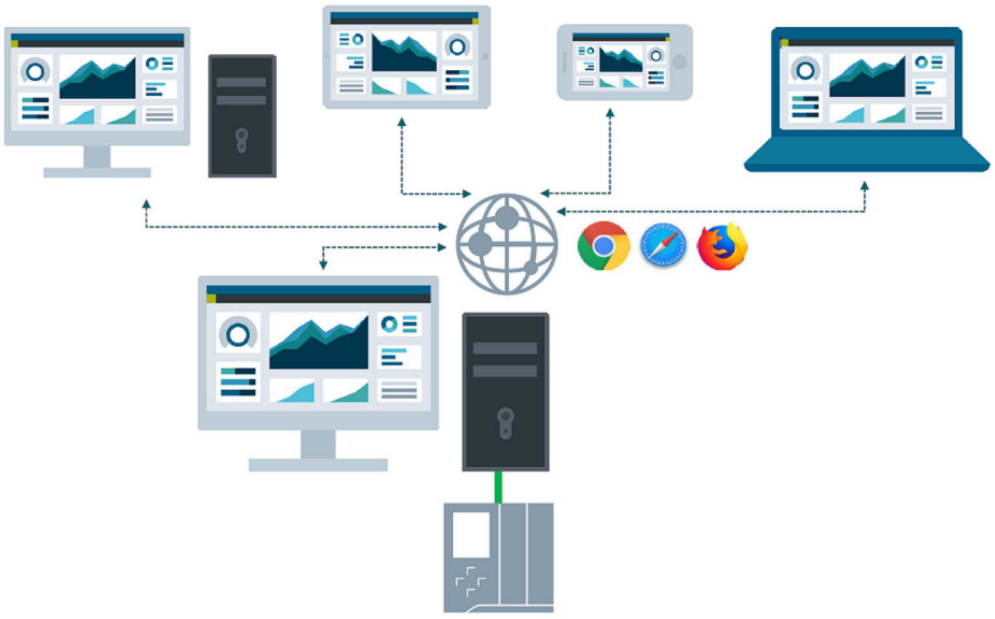
Nowadays, in order to increase productivity, reduce costs and improve product quality, automation is an inevitable process in almost all industries. However, the successful application of automation in complex industrial systems requires a high-quality human-machine interface that enables easier transfer of information between the system and the operator, thus solving a numerous automation development problems. For this reason, it was decided to develop an industrial user interface for monitoring and managing industrial and collaborative robots and transportation systems in the Laboratory for Autonomous Systems within the Regional Center of Excellence for Robotic Technologies (CRTA). The aim of the thesis is to show the process of creating such a user interface using the Siemens "SIMATIC WinCC Unified" and to demonstrate the advantages of its use. In addition to the creation of the user interface, the creation of the corresponding PLC program and function blocks was also presented in order to standardize system parts so that they can be more easily used in modular form in the future.

Key words: automation, industrial user interface, CRTA, SIMATIC WinCC Unified, PLC, standardization, modular use

1. UVOD

U 21. stoljeću, roboti su postali neizostavan dio gotovo svih industrijskih procesa. Osim što se njihovim korištenjem povećava učinkovitost i produktivnost sustava, ujedno se smanjuje ljudski faktor što proporcionalno smanjuje rizik od pogreške te povećava sigurnost rada operatera. Roboti svojom fleksibilnošću omogućuju operaterima rješavanje niza zadataka koji su se prethodno smatrali teškima i opasnima, a danas su postali svakodnevnica svakog postrojenja. Za izvođenje kompleksnih zadataka s velikom preciznošću, roboti su nerijetko povezani s PLC-om (eng. *Programmable Logic Controller*) koji uvelike pridonosi stabilnosti i ponovljivosti rada automatiziranih sustava. PLC je direktno odgovoran za kontrolu i nadzor robota, za upravljanje jednostavnijim elementima sustava poput senzora i aktuatora te za obrađivanje i skladištenje svih podataka koji se izmjenjuju unutar sustava. Kako bi se u što većoj mjeri iskoristila već spomenuta fleksibilnost robotskih sustava, od velike je važnosti uz PLC imati i kvalitetno korisničko sučelje koje će omogućiti operateru direktnu komunikaciju sa sustavom, a time i upravljanje i nadzor svih njegovih komponenti. Korisničko sučelje mora biti intuitivno i lako shvatljivo za svakog operatera, što nije uvijek jednostavno postići zbog kompleksnosti nekih proizvodnih procesa, primijenjenih tehnologija proizvodnje, ali i potrebe za automatiziranom kontrolom kvalitete svih proizvoda.

U tu se svrhu tvrtka Siemens, potaknuta četvrtom industrijskom revolucijom, odlučila na razvoj novog alata pod nazivom „SIMATIC WinCC Unified“ koji omogućuje stvaranje modernih korisničkih sučelja imajući na umu potrebu za unaprjeđenje tradicionalnih korisničkih sučelja u smislu njihove funkcionalnosti, ali i vizualizacije. Za razliku od njihovih prijašnjih SIMATIC HMI proizvoda koji su bili bazirani na zastarjelom Visual Basic programskom jeziku, WinCC Unified koristi JavaScript programski jezik koji donosi niz prednosti. JavaScript je *open-source* jezik specijaliziran za izradu web aplikacija koji je ujedno uspješno integriran s ostalim web tehnologijama poput HTML-a, CSS-a, JSON-a i slično [1]. Samim time Siemens korištenjem JavaScript-a kao programskog jezika HMI (eng. *Human Machine Interface*) aplikacija nove generacije postiže direktan cilj Industrije 4.0, a to je pametna proizvodnja s mogućnošću nadzora s udaljenog mjesta i to s vrlo jednostavnim pristupom, odnosno korištenjem samo web preglednika [Slika 1].



Slika 1. Mogućnost pristupa WinCC Unified korisničkom sučelju preko bilo kojeg uređaja s web preglednikom [1]

1.1. Proizvodni sustavi u industriji

U industriji, tradicionalna automatizacija postiže se rastavljanjem proizvodnog sustava na više modula, pri čemu je svaki modul nezavisan o drugima u smislu programiranja, nadzora i upravljanja. Takvi moduli obično iznad sebe posjeduju MES (eng. *Manufacturing Execution System*) sustav koji se brine o upravljanju proizvodnim sustavima na razini proizvodnog pogona. Međutim, za unošenje promjena na određenim dijelovima proizvodnog sustava često je potrebno imati operatera koji je fizički na lokaciji te koji mora ručno unositi željene promjene zasebno na svakom modulu s obzirom da moduli ne nude mogućnost centralnog upravljanja.

Slika 2. Primjer automatizirane proizvodne linije u industriji [2]

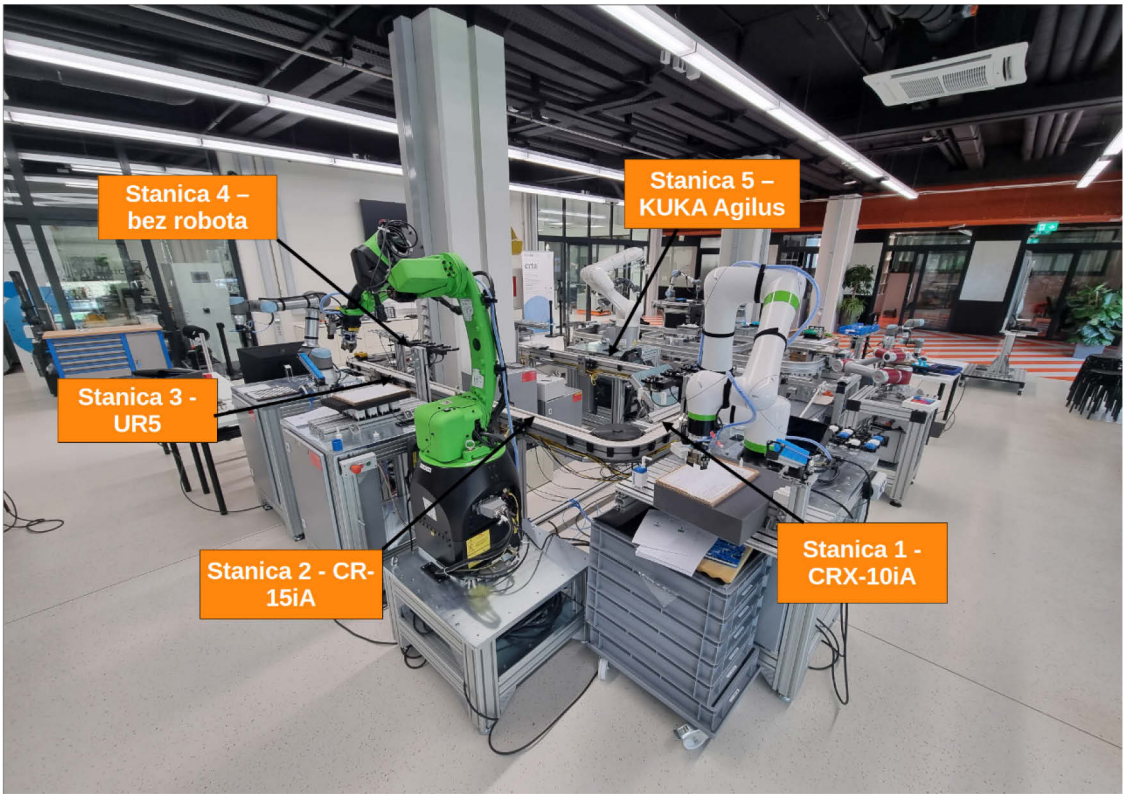
Takav zastarjeli način upravljanja uvelike povećava mogućnost pogreške operatera što u krajnjem slučaju može uzrokovati velike štete u obliku izgubljenog vremena ili stvaranju škarta. Upravo iz tog razloga koriste se korisnička sučelja nove generacije, koja omogućuju jednostavnije unošenje promjena u sustav, ali i jednostavniji i pregledniji nadzor nad unesenim promjenama. Takva sučelja u kombinaciji s MES sustavima rezultiraju znatnim povećanjem učinkovitosti proizvodnje i povećanju profitabilnosti tvrtke te je stoga od iznimne važnosti težiti prema razvoju i testiranju takvih sustava. Unutar ovog diplomskog rada uspostavljeno je takvo korisničko sučelje na proizvodnoj liniji koja vjerno replicira proizvodnu liniju modernog postrojenja.

2. OPIS KOMPONENTI ROBOTSKOG SUSTAVA

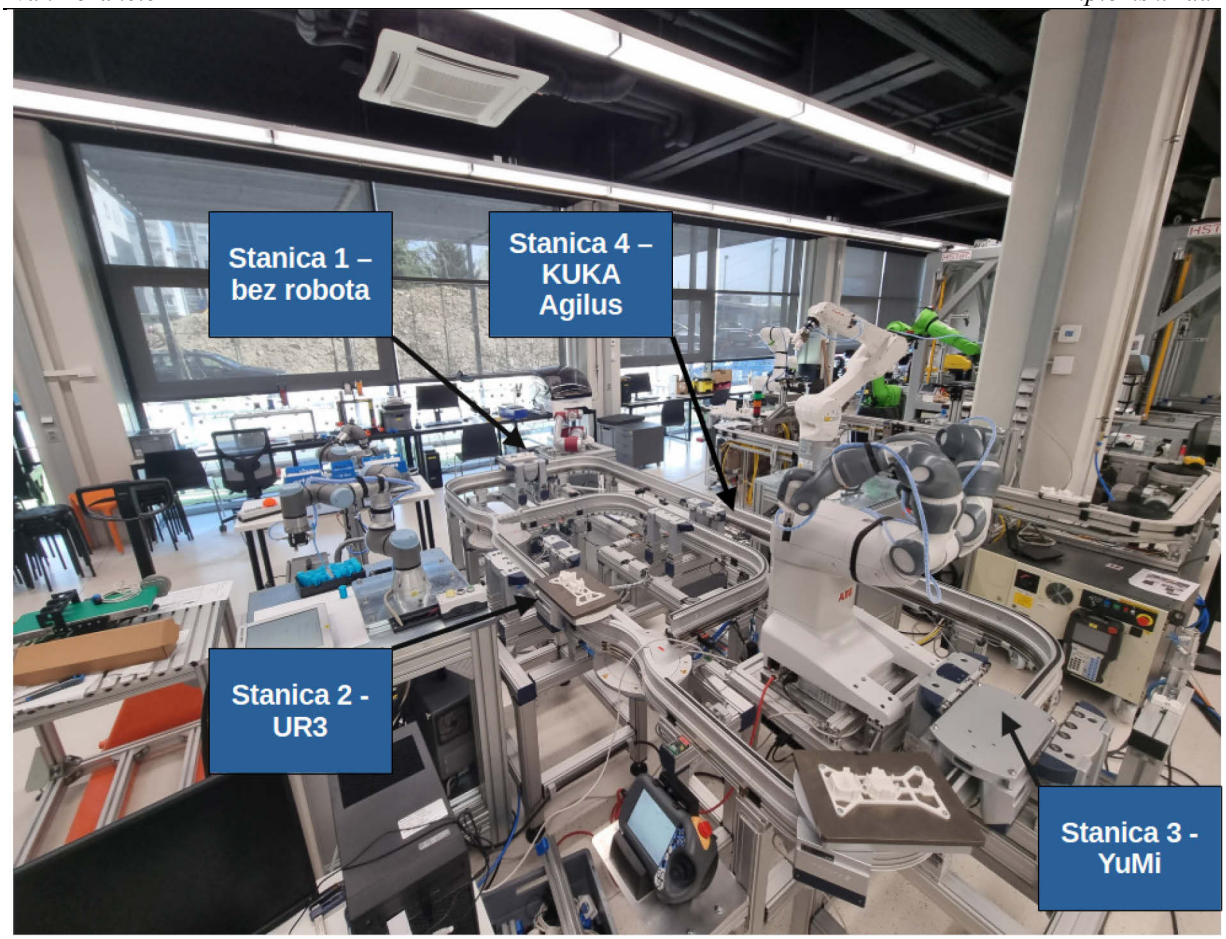
Proizvodni sustav za koji je u sklopu ovog diplomskog rada izrađeno korisničko sučelje nalazi se u Regionalnom centru izvrsnosti za robotske tehnologije, poznatijeg pod kraticom CRTA, točnije unutar Laboratorija za autonomne sustave. Navedena proizvodna linija ima funkciju sastavljanja 3D printanog modela FSB-ovog robota razvijenog u sklopu projekta RONNA (Robotska neuronavigacija).

Proizvodna se linija može podijeliti u dva modula, gdje je svaki modul organiziran oko svog transportnog sustava. Transportni su sustavi od različitih proizvođača te su zbog jednostavnosti nazivlja moduli nazvani prema njima. Prema tome razlikujemo:

- Rexroth modul [Slika 3], prema transportnom sustavu VarioFlow plus, proizvodu tvrtke Bosch Rexroth
- Montrac modul [Slika 4], prema transportnom sustavu Montrac, proizvodu tvrtke Montratec

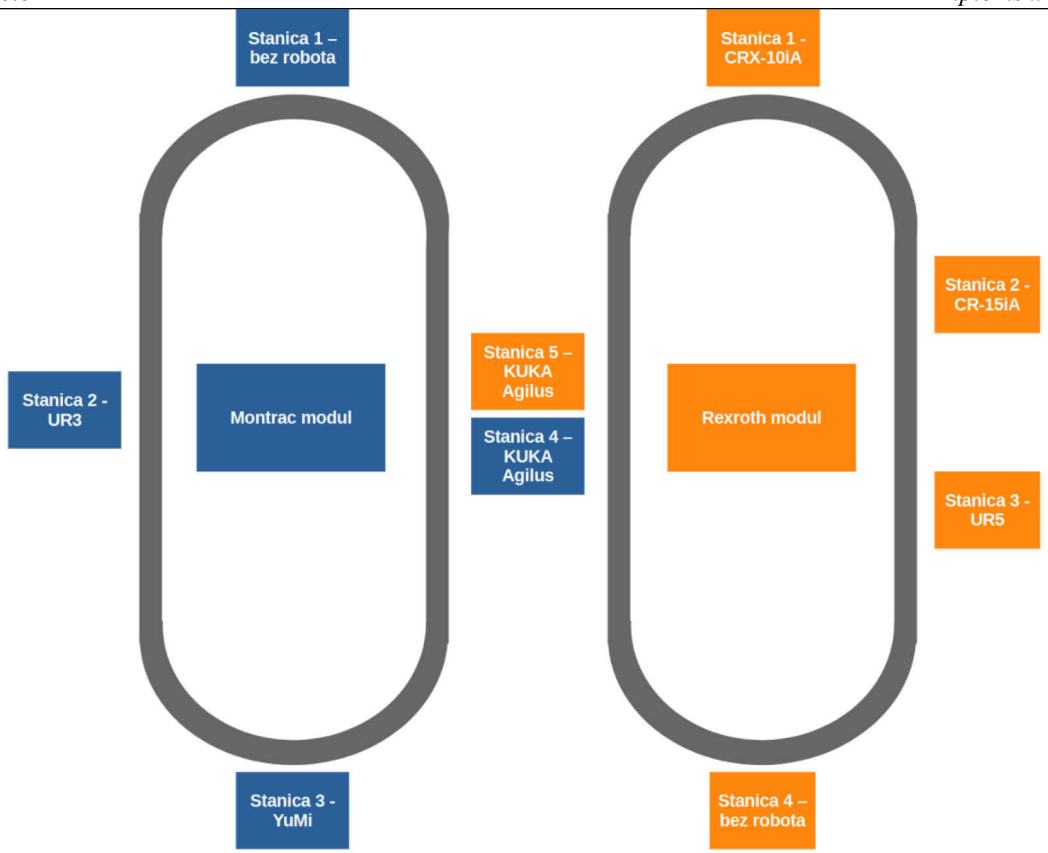


Slika 3. Rexroth modul s označenim proizvodnim stanicama



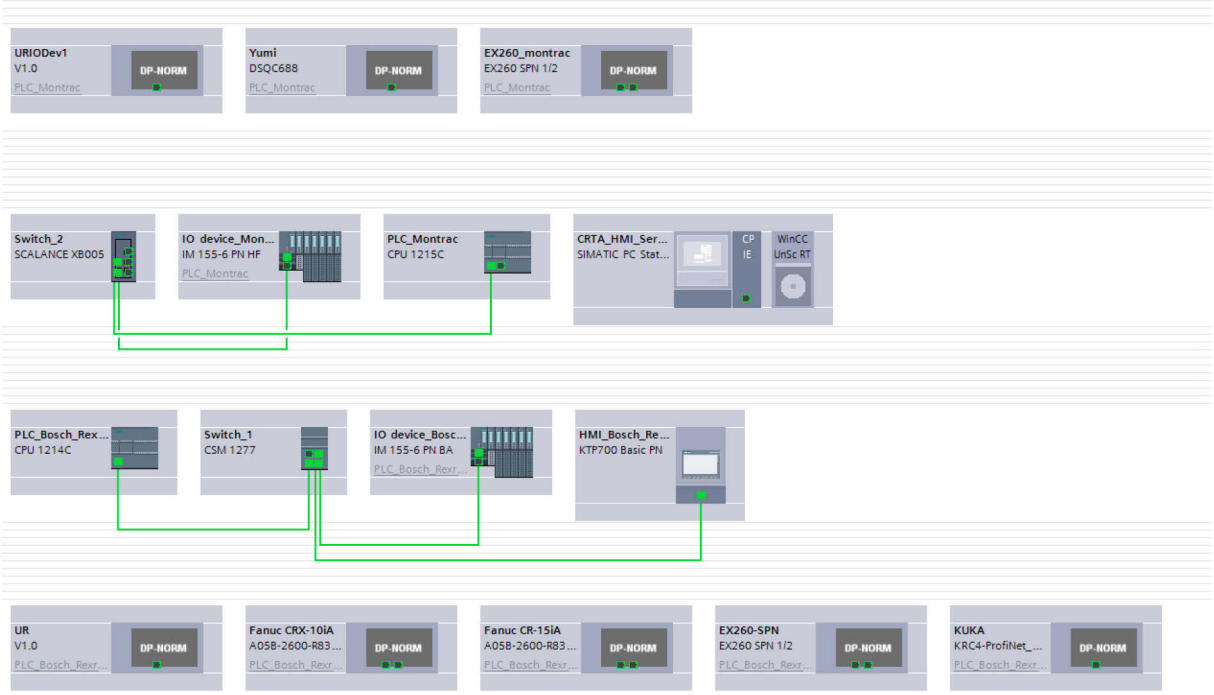
Slika 4. Montrac modul s označenim proizvodnim stanicama

U sustavu se sveukupno nalazi šest robota, od kojih su tri pozicionirani po stanicama Rexroth modula, dva pozicionirana po stanicama Montrac modula te jedan robot koji služi kao robot za transfer materijala, odnosno prijenos proizvodnih dijelova s jednog transportnog sustava na drugi. Radi lakšeg razumijevanja, opisani se sustav može prikazati pomoću blokovskog dijagrama [Slika 5]. Oznake proizvodnih stanica na dijagramu odgovaraju onima unutar modula, pa tako KUKA Agilus robot se nalazi pod stanicom 4 unutar Montrac modula te pod stanicom 5 unutar Rexroth modula.



Slika 5. Blokovski dijagram proizvodnog sustava s označenim proizvodnim stanicama

Od navedenih šest robota u sustavu, pet je kolaborativnih robota koji su u mogućnosti dijeliti svoj radni prostor s ljudima, dok je samo jedan industrijski robot čiji radni prostor radi sigurnosti mora biti striktno odvojen i van ljudskog kontakta te koji je upravo iskorišten kao robot za transfer materijala gdje nema potrebe za ljudskom interakcijom. Svaki modul ima zaseban PLC koji se brine za nadzor i kontrolu proizvodnog procesa vezanog za svoj transportni sustav. Osim PLC-a, Rexroth modul ima i jedan fizički HMI koji je služio za kontrolu Rexroth transportnog sustava. Kako je cilj diplomskog rada napraviti unaprjeđenje sustava mobilnim HMI sučeljem, statički HMI gubi svoj smisao. Međutim, njegov rad nije sputavan unaprjeđenjem sustava te se i dalje može koristiti za testiranje rada transportnog sustava Rexroth modula. Na [Slika 6] vidljiv je topološki prikaz svih komponenti sustava, kao i prikaz fizičke veze između PLC-a i njegovih popratnih modula.



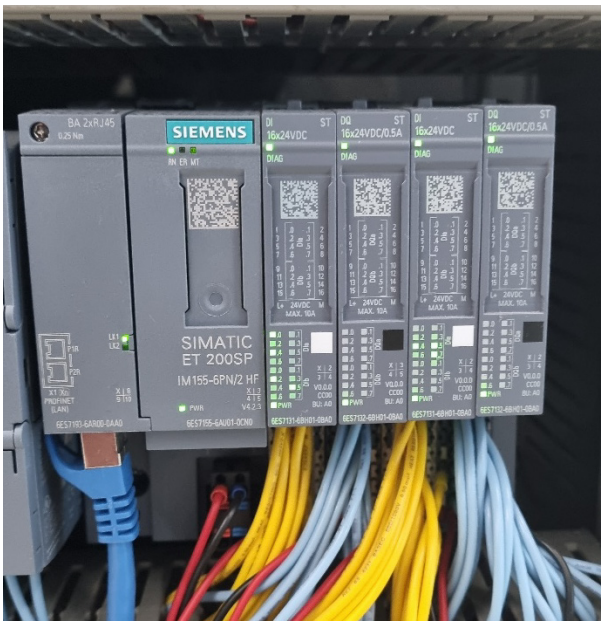
Slika 6. Topološki prikaz cijelog sustava unutar TIA Portal projekta

2.1. PLC i popratni moduli

Montrac modul koristi Siemensov kontroler oznake 1215C DC/DC/DC [Slika 7]. DC (eng. *Direct current*) označava izvor napajanja, gdje prvi DC simbol označava napajanje kontrolera, a druga dva simbola opisuju napajanje fizičkih ulaza i izlaza smještenih na kontroleru. PLC kontroler na sebi ima 14 digitalnih ulaza, 10 digitalnih izlaza te po dva analogna ulaza i izlaza. Važno je napomenuti da PLC također podržava OPC UA (eng. *Open Platform Communications Unified Architecture*) protokol koji je nužan za integraciju i komunikaciju WinCC Unified sustava s PLC-om.

Slika 7. Prikaz PLC-a 1215C DC/DC/DC i SCALANCE XB005 *switch*-a na Montrac modulu

Zbog potrebe za dodatnim fizičkim ulazima i izlazima za kontrolu Montrac transportnog sustava, na PLC je spojen modularni sustav SIMATIC ET200SP [Slika 8] koji omogućuje nadogradnju sustava s dodatnim ulaznim i izlaznim karticama te je tako za potrebe Montrac modula dodano još dvije ulazne kartice s 16 ulaza i dvije izlazne kartice s 16 izlaza.



Slika 8. Modularni sustav SIMATIC ET200SP s po dvije ulazne i izlazne kartice na Montrac modulu

Na postojeći sustav je zbog potrebe za kontrolom ventila dodan i elektronički ventilski blok EX260 [Slika 9] koji omogućuje kontrolu ventilima preko PROFINET-a. ET200SP i EX260 su

s PLC-om spojeni pomoću SCALANCE XB005 neupravljivog industrijskog mrežnog preklopnika (eng. *switch*).

Slika 9. Elektronički ventilski blok EX260 SPN 1/2 na Montrac modulu

Rexroth modul koristi vrlo sličnu opremu kao i Montrac, gdje je glavna razlika što se u Rexrothovom modulu koristi PLC oznake 1214C DC/DC/DC. Ukoliko se usporedi tehnička dokumentacija ova dva PLC-a, vidljivo je da je glavna razlika u tome što 1215C ima na sebi dva PROFINET priključka dok 1214C ima samo jedan. Modul također koristi ET200SP s dvije ulazne i dvije izlazne kartice te ventilski blok EX260, međutim ovdje su oni spojeni s PLC-om preko CSM 1277 neupravljivog industrijskog *switch*-a. Kao što je već spomenuto, kod Rexrothovog modula se također nalazi fizički HMI oznake KTP700 Basic.

2.2. Roboti

Na Montrac modulu nalaze se dva kolaborativna robota, YuMi [Slika 10] i UR3 [Slika 11]. YuMi je skraćeni naziv za robota IRB 14000 razvijenog od strane tvrtke ABB, a naziv je dobio po sloganu *„You and me*“, odnosno „*Ti i ja*“ što se odnosi na njegove kolaborativne sposobnosti. YuMi je poznat po tome što ima dvije robotske ruke, od kojih svaka ima sedam osi, što ga čini najagilnijim kolaborativnim robotom dosad [3]. Robot je razvijen imajući na umu proizvodne procese koji zahtijevaju visoku preciznost i sklapanje proizvoda malih dimenzija. Njegov impresivni ugrađeni sustav sigurnosti čini ga iznimno sigurnim za bilo kakve kolaborativne primjene što omogućuje najveću sigurnu radnu brzinu od svih kolaborativnih robota [3].

Slika 10. ABB YuMi robot

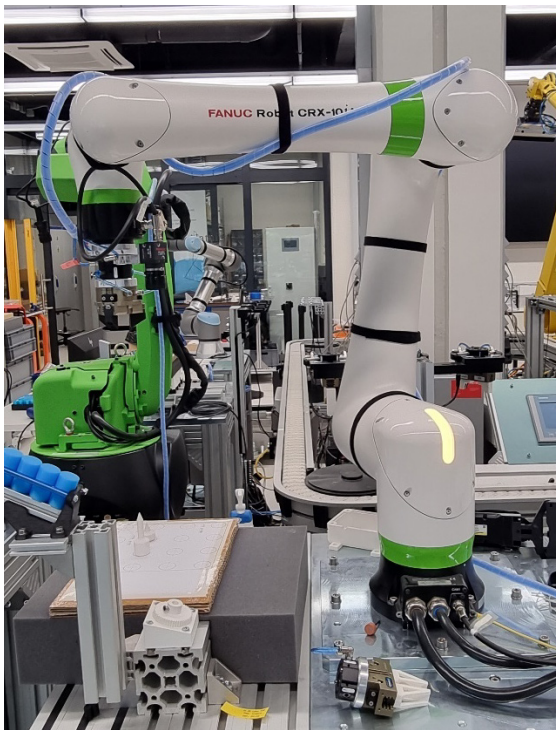
UR3 je najmanji i najlakši robot tvrtke Universal Robots. Robot mase od samo 11 kilograma ima impresivnu nosivost od tri kilograma. S ponovljivošću od 0,1 milimetra i s rotacijom od 360 stupnjeva na svim zglobovima te senzorima sile koji pružaju sigurnost pri radu, UR3 postaje odličan izbor za automatizaciju gdje može služiti kao pomoć operateru pri izvođenju zahtjevnijih operacija [4]. UR roboti se mogu programirati ručnim vođenjem robota, što uvelike skraćuje proces programiranja robota.

Slika 11. UR3 robot

Rexroth modul za obavljanje proizvodnog procesa koristi tri različita kolaborativna robota, CR-15iA [Slika 12], CRX-10iA [Slika 13] i UR5 [Slika 14]. CR i CRX serije su kolaborativnih

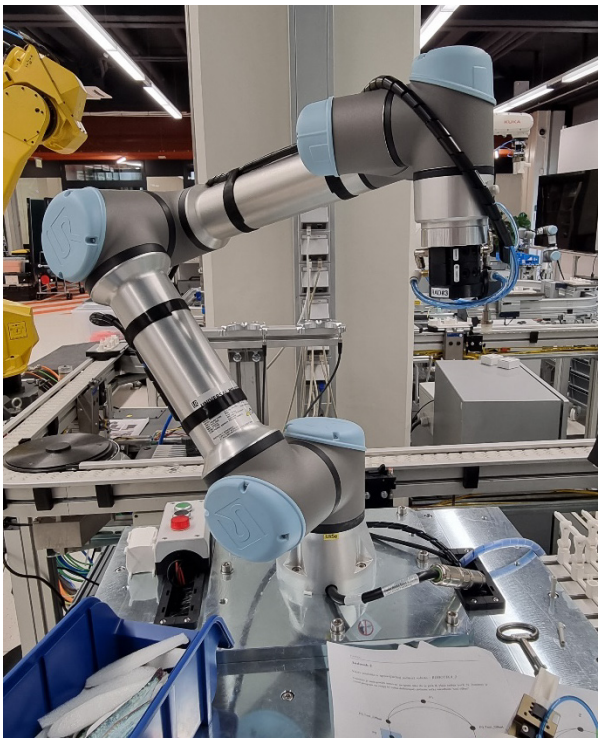
robota tvrtke Fanuc te imaju vrlo sličnu primjenu u industriji. Kod obje serije velika je pozornost posvećena na sigurnost, svi su roboti glatkih površina bez oštrih kutova te su opremljeni kontaktnim senzorima za izbjegavanje kolizije [5]. Kao i kod UR robota, ove robote je također moguće pomicati ručnim vođenjem. Iako veoma slični po primjeni, vrlo ih je lako raspoznati po boji kućišta: CR roboti su zelene, dok su CRX roboti bijele boje.

Slika 12. Fanuc CR-15iA robot



Slika 13. Fanuc CRX-10iA robot

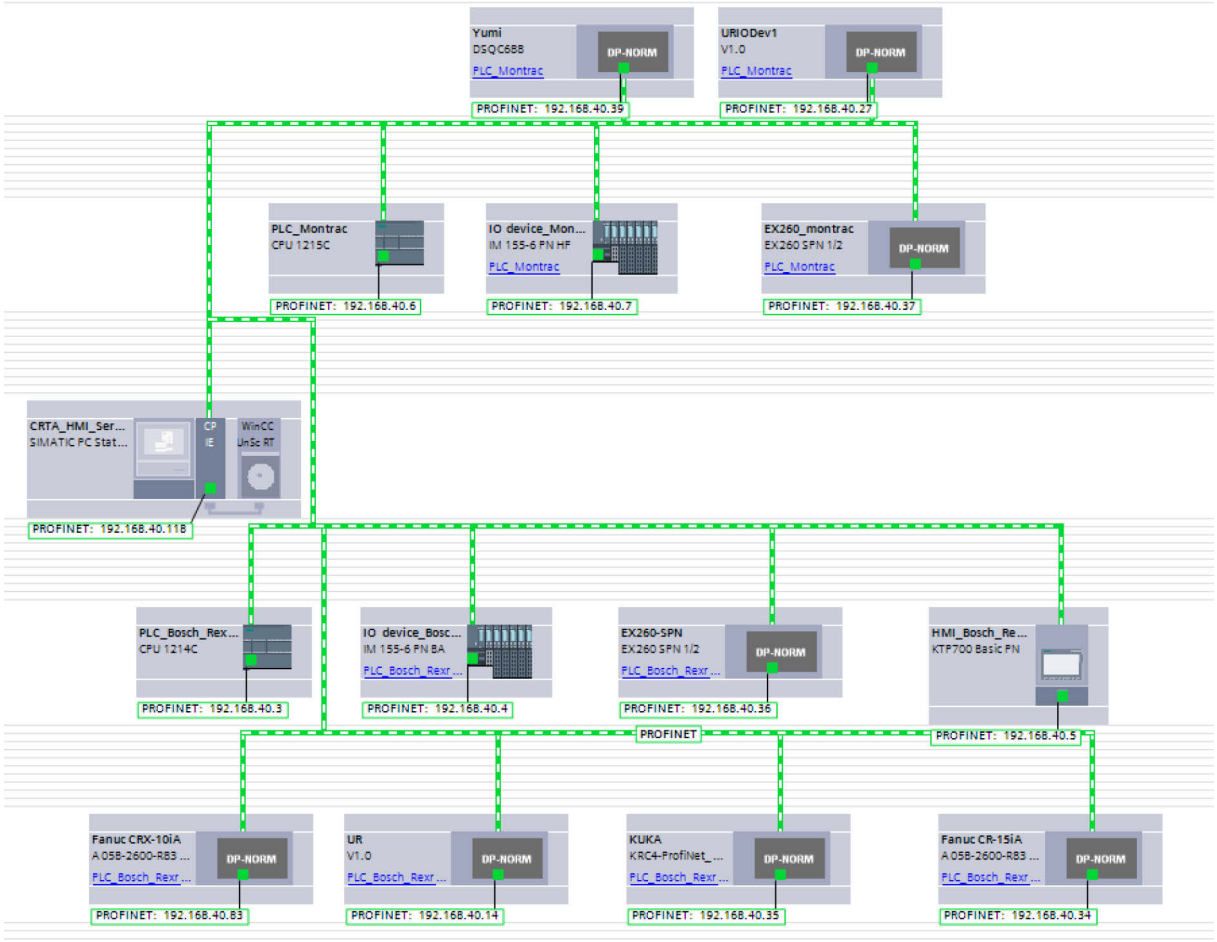
UR5 je, kako izgledom, a tako i funkcijom, veoma sličan prethodno opisanom UR3 robotu, gdje su glavne razlike u samo većoj nosivosti i dosegu.



Slika 14. UR5 robot

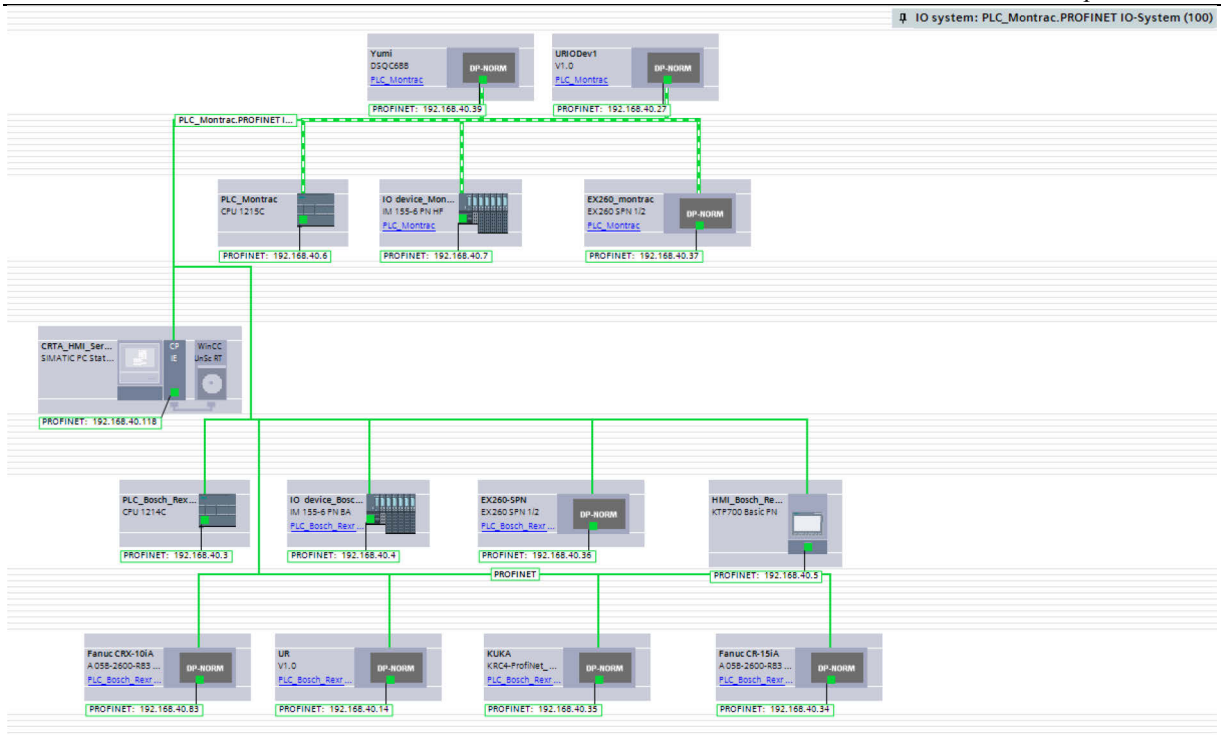
2.3. Komunikacija između komponenti

Pošto su u laboratoriju sve opisane komponente na istoj podmreži (eng. *subnet*), potrebno ih je tako i definirati u TIA Portal-u. Za postizanje toga, potrebno je bilo postojeće projekte za Montrac i Rexroth modul spojiti u jedan TIA Portal projekt te korektno definirati fizički postojeću podmrežu. Potom je potrebno bilo dodati PC stanicu na kojoj se nalazi WinCC Unified koja također mora biti u istoj podmreži sa svim komponentama unutar projekta [Slika 15].



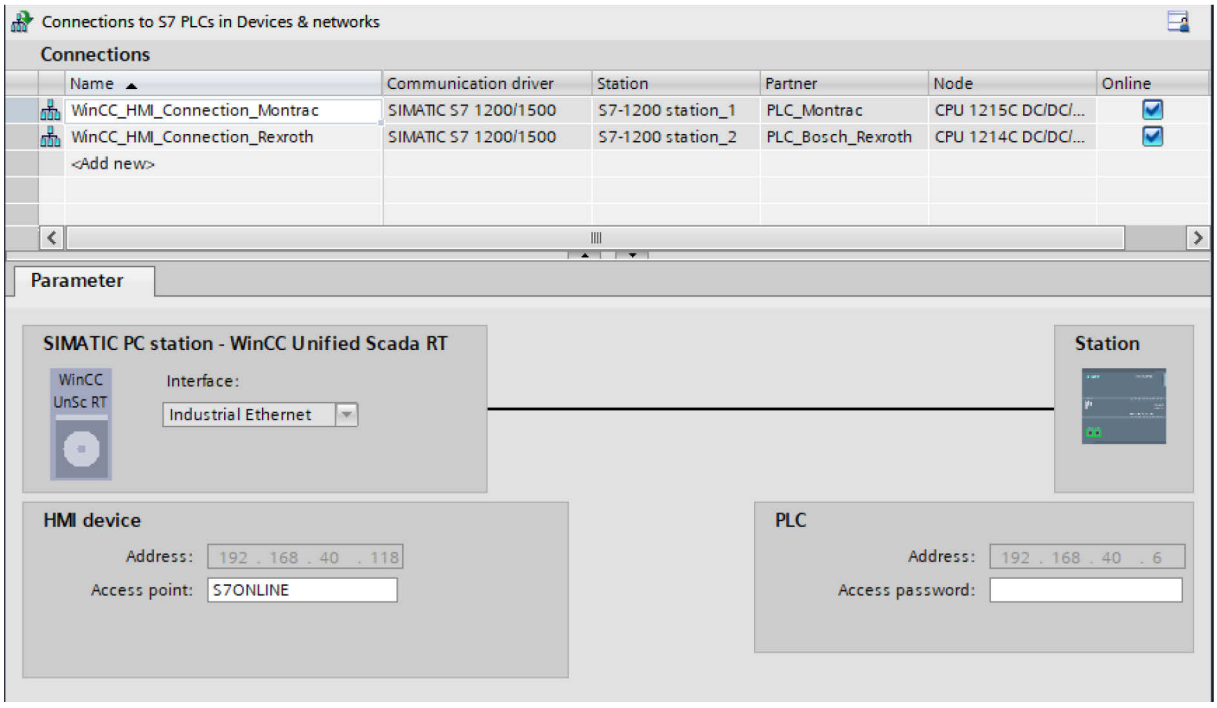
Slika 15. Prikaz definirane podmreže u TIA Portal projektu

Iako su svi uređaji na istoj podmreži, u stvarnosti i dalje imamo dva različita PROFINET IO sustava za dva neovisna modula gdje je svaki PLC IO kontroler, a sve ostale komponente unutar modula IO uređaji [Slika 16].



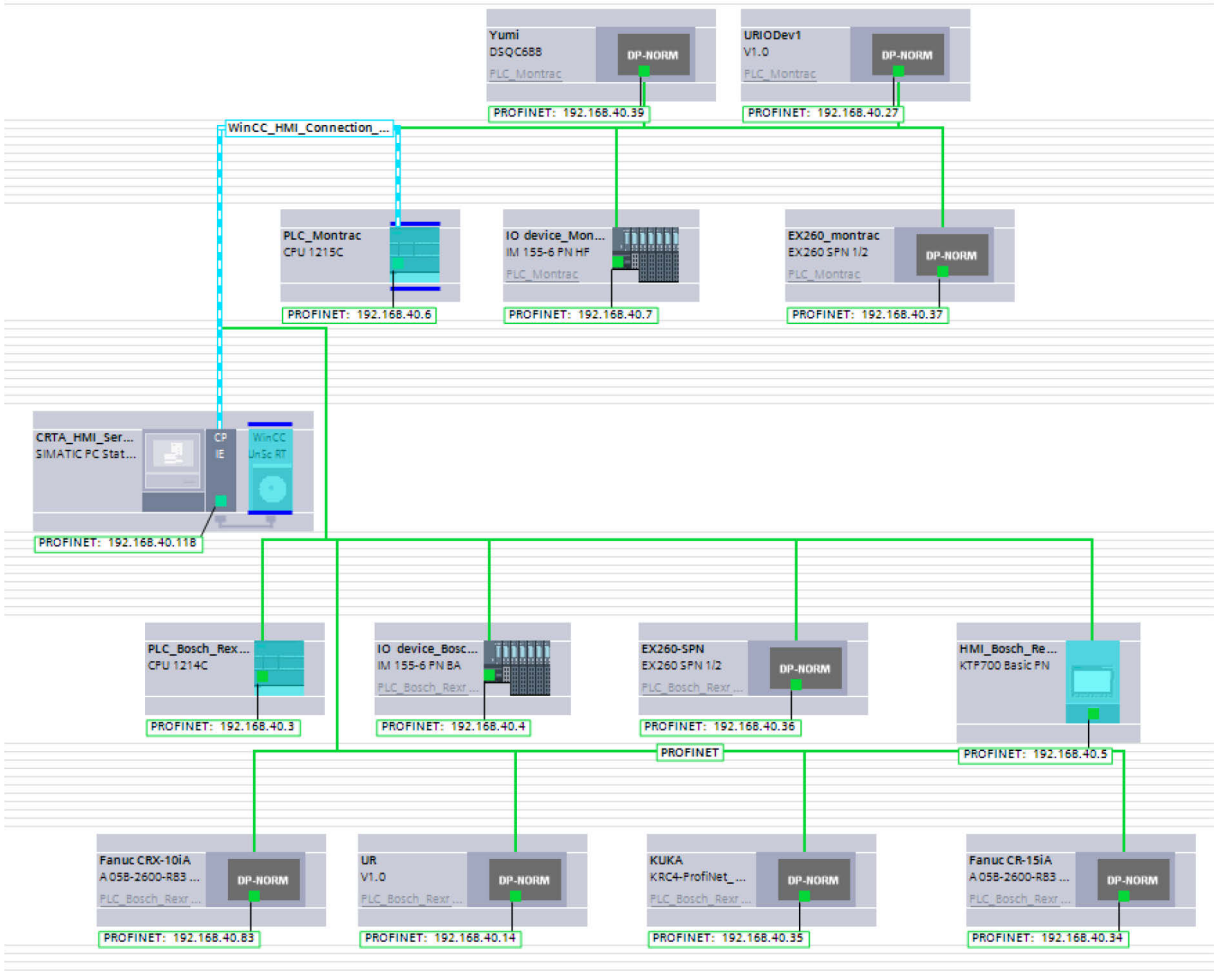
Slika 16. Prikaz Montrac PROFINET IO sustava u TIA Portal projektu

Osim definiranja podmreže, za korištenje HMI sučelja potrebno je definirati i HMI veze između IO kontrolera i HMI uređaja. Na već jednu postojeću HMI vezu između fizičkog HMI-a i Rexroth PLC-a dodane su još dvije HMI veze, jedna između Rexroth PLC-a i WinCC Unified PC-a kao i jedna između Montrac PLC-a i WinCC Unified PC-a [Slika 17].



Slika 17. Prikaz definiranih HMI veza između WinCC Unified PC-a i PLC-a

Sve postojeće HMI veze moguće je vidjeti na zaslonu TIA Portala „Network view“, gdje će uređaji koji imaju HMI vezu biti označeni svijetloplavom bojom te gdje će se svaka HMI veza moći posebno naglasiti [Slika 18].



Slika 18. Naglašeni prikaz HMI veze između WinCC Unified PC-a i Montrac PLC-a u TIA Portal projektu

U Laboratoriju je već prethodno uspostavljena komunikacija između dva PLC-a pomoću funkcijskih blokova TSEND_C i TRCV_C koji omogućuju jednostavnu PLC-PLC komunikaciju.

3. IZRADA I MODIFIKACIJA PROGRAMSKE PODRŠKE SUSTAVA

3.1. Programska podrška PLC-a

Za programiranje PLC-a korišten je programski paket TIA Portal punog naziva „Totally Integrated Automation Portal*"*. TIA Portal objedinjuje sve SIMATIC TIA proizvode tvrtke Siemens u jedinstvenu programsku aplikaciju koja se koristi za konfiguraciju i programiranje PLC-a, HMI-a, IO uređaja, njihovih odnosa u obliku mrežnih veza i slično. TIA Portal ima različit broj verzija koje su međusobno kompatibilne prema nižim verzijama. Međutim, treba naglasiti da projekt napravljen pomoću TIA Portala više verzije nije moguće nikako više otvoriti pomoću niže verzije. Prilikom izrade ovog diplomskog rada korištena je isključivo TIA Portal V16 verzija.

TIA Portal V16 za programiranje PLC-a 1200 serije omogućuje korištenje četiri različita programska jezika:

1. STL (Statement List) – tekstualni programski jezik nalik asembleru
2. LAD (Ladder Diagram) – grafički programski jezik temeljen na shematskom prikazu električnih sklopova
3. FBD (Function Block Diagram) – grafički programski jezik koji koristi blokovski prikaz logike
4. SCL (Structured Control Language) – tekstualni programski jezik više razine od STL-a, temeljen na sintaksi programskoj jezika Pascal

Potrebno je napomenuti kako su LAD i FBD jezici međusobno potpuno zamjenjivi te je promjena programskog jezika prilikom programiranja vrlo jednostavna. Radi preglednosti, prilikom izrade diplomskog rada prednost je dana FBD programskom jeziku. Osim izgledu programske podrške na osnovi korištenog programskog jezika, posvećena je pozornost i na strukturu PLC programa, prvenstveno kako bi se budući PLC programeri lakše snašli unutar projekta te samim time lakše i brže unaprjeđivali sustav. Svaki PLC ima glavni organizacijski blok (Main OB) koji se ciklički izvršava na početku rada PLC-a prije bilo kojeg drugog funkcijskog bloka. Main OB služi kao inicijalizacija ostalih blokova te će se u ciklusnom radu PLC-a izvršiti samo oni funkcijski blokovi koji su bili inicijalizirani.

3.1.1. Glavni organizacijski blok

Struktura Main OB-a PLC-a je podijeljena u četiri odjeljka [Slika 19]:

1. Stanice
2. Transport
3. Komunikacija
4. Sigurnost

Unutar odjeljka „Stanice“ nalaze se funkcijski blokovi čiji nazivi odgovaraju rednom broju stanice, odnosno svaka stanica na modulu ima svoj funkcijski blok unutar kojeg je sadržana sva logika za tu stanicu. Time se postiže pregledan prikaz Main OB-a, ali i lakše snalaženje u projektnom stablu prilikom potrebe za izmjenom određene stanice. Pod odjeljkom „Transport“ nalaze se funkcijski blokovi koji direktno upravljaju transportnim sustavom, odnosno konvejerom. Pod odjeljkom „Komunikacija“ nalaze se funkcijski blokovi koji omogućuju konstantnu PLC-PLC komunikaciju i međusobnu razmjenu podataka. Pod odjeljkom „Sigurnost“ nalaze se svi funkcijski blokovi odgovorni za sigurnost cijelog sustava te za svjetlonosnu vizualizaciju stanja istog pomoću industrijskog semafora. Jedina razlika između Main OB-a Rexroth i Montrac modula je ta što Montrac modul nema odjeljak „Sigurnost“, pošto je sva logika već riješena preko PLC-a u Rexroth modulu.

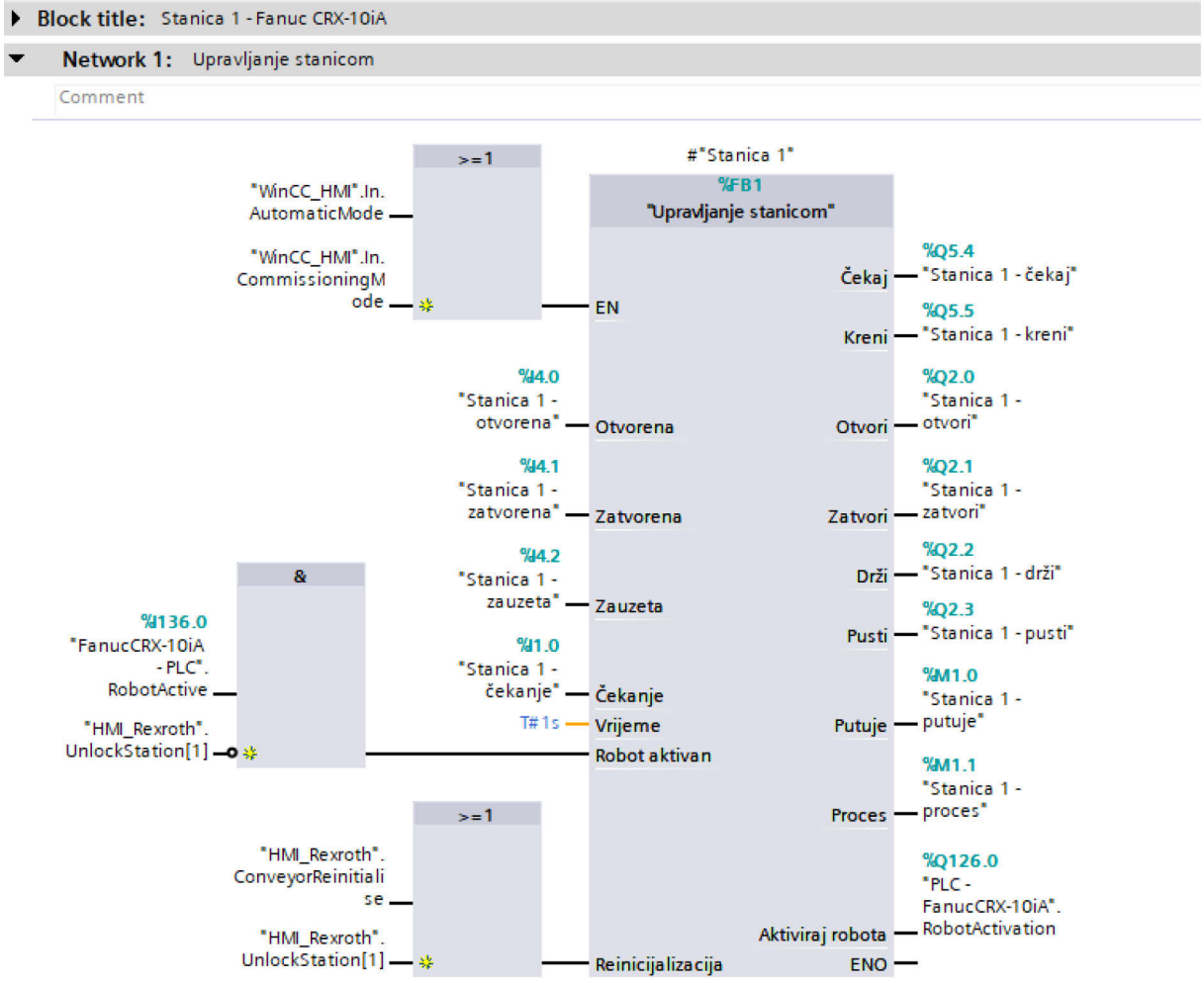
Slika 19. Struktura Main OB-a Rexroth PLC-a u TIA Portal projektu

3.1.1.1. Stanice

Struktura funkcijskog bloka stanice je također unificirana između svih stanica te je logika podijeljena na četiri glavne mreže:

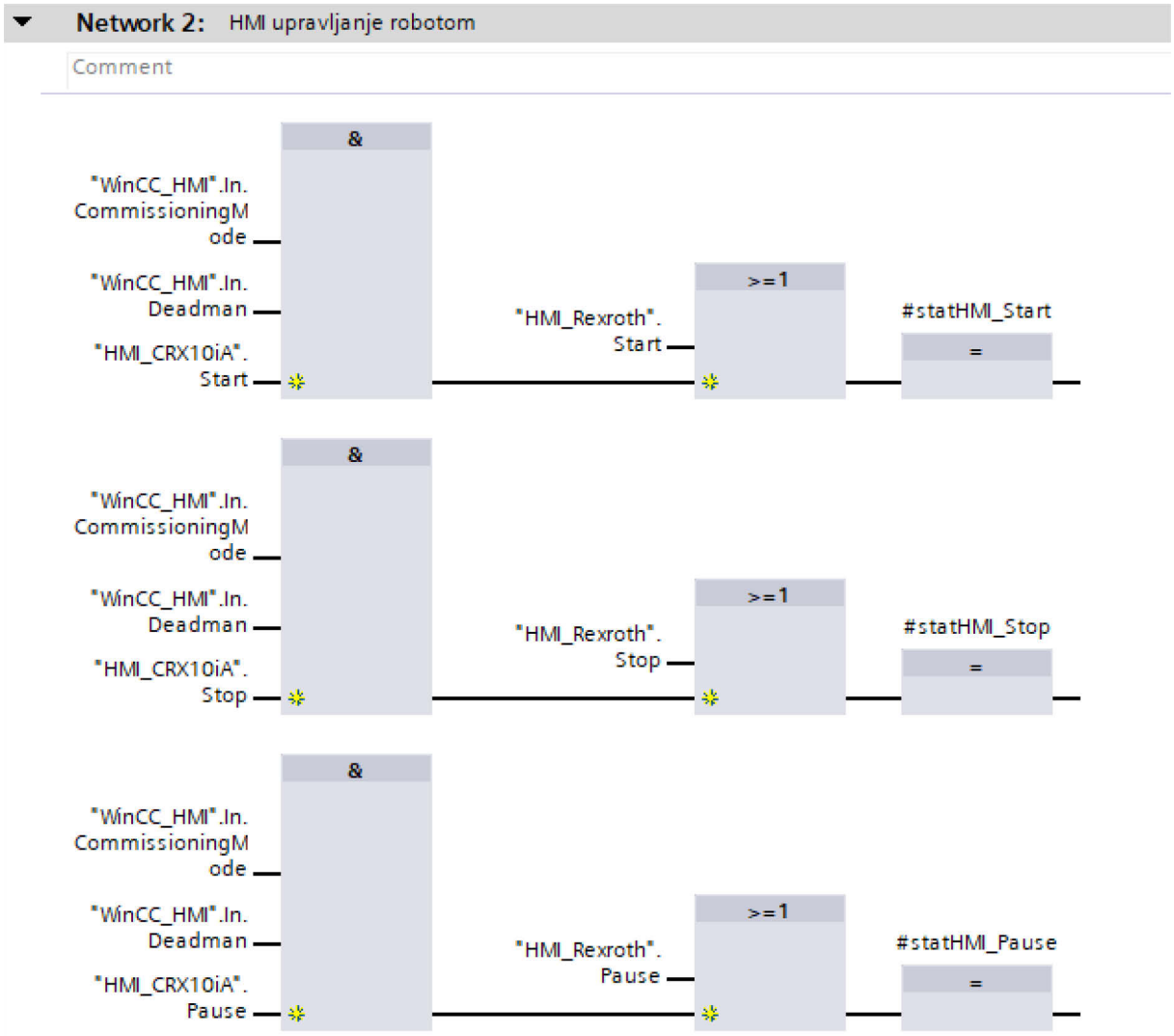
1. Upravljanje stanicom
2. HMI upravljanje robotom
3. Robot
4. Upravljanje hvataljkom

Unutar mreže „Upravljanje stanicom“ [Slika 20] nalazi se već prethodno definirana logika za kontrolu prvenstveno zaključavanja i otključavanja pozicije palete s proizvodom. Kada paleta dođe u radni položaj, njeno kretanje se ograniči pomoću cilindara kako bi osigurali točnost i ponovljivost rada robota. Nakon osiguravanja pozicije palete, šalje se signal robotu koji omogućuje izvršavanje procesa sklapanja proizvoda. Nakon izvršenja procesa, robot šalje signal koji omogućuje automatsko otključavanje palete te njeno kretanje prema sljedećoj stanici. Logika rada opisanog funkcijskog bloka nije mijenjana, već je samo nadograđena s dodatnim ulaznim signalima za reinicijalizaciju stanice te za omogućavanje rada funkcijskog bloka ovisno o načinu rada sustava, koji se definira pomoću HMI-a.



Slika 20. Mreža za upravljanje stanicom 1 na Rexroth modulu

Unutar mreže „HMI upravljanje robotom“ [Slika 21] nalazi se logika koja omogućava rad robota u automatskom načinu rada kao i u načinu puštanja pogona u rad. Takva logika je nužna za individualno testiranje rada svakog robota.



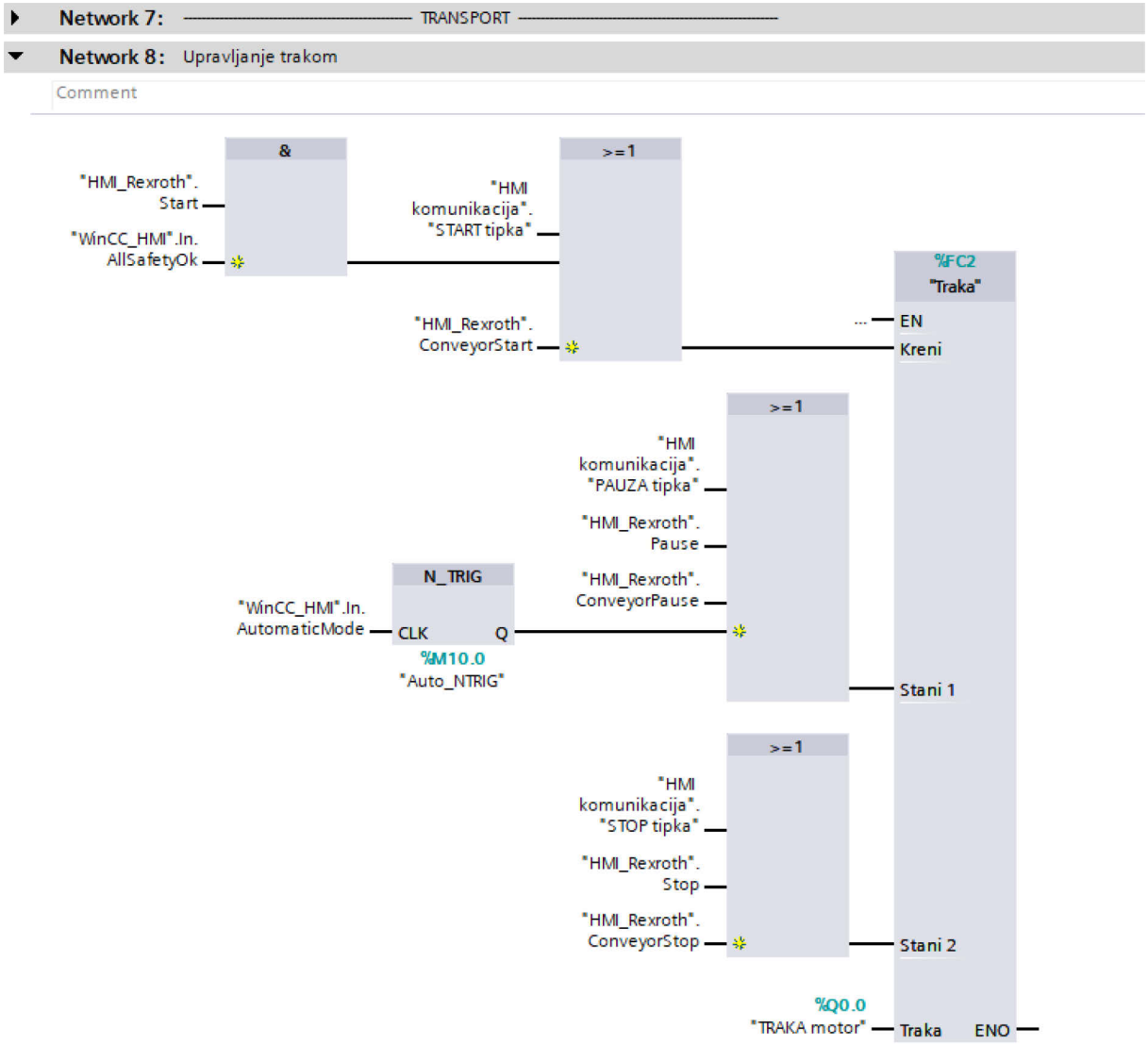
Slika 21. Mreža za HMI upravljanje robotom ovisno o načinu rada sustava unutar funkcijskog bloka stanice

Unutar mreže „Robot“ nalazi se funkcijski blok koji se brine o svoj logici vezanoj za rad robota. Funkcijski blokovi su standardizirani prema uputama proizvođača robota, te se funkcijski blok jednog proizvođača teoretski može koristiti za sve njihove robote koji koriste isti proces inicijalizacije i komunikacije s PLC-om. Kao ulazne signale za ovakve funkcijske blokove koriste se izlazni signali robota te signali vezani za upravljanje preko HMI-a. Rad funkcijskog bloka robota bit će detaljno opisan u sljedećim poglavljima.

Unutar mreže „Upravljanje hvataljkom“ nalazi se logika potrebna za omogućavanje kontrole hvataljke robota pomoću HMI-a, ali i logika za direktnu kontrolu hvataljke robotom, neovisno o ostatku sustava. Logika se razlikuje ovisno o robotu te o vrsti hvataljke.

3.1.1.2. Transport

Transportni sustav Rexroth modula dosad je bio upravljan pomoću statičkog HMI-a. Funkcija za upravljanje transportnim sustavom nije mijenjanja, već su na postojeće ulazne signale funkcije dodani dodatni signali koji će također omogućiti kontrolu pomoću novog korisničkog sučelja [Slika 22].



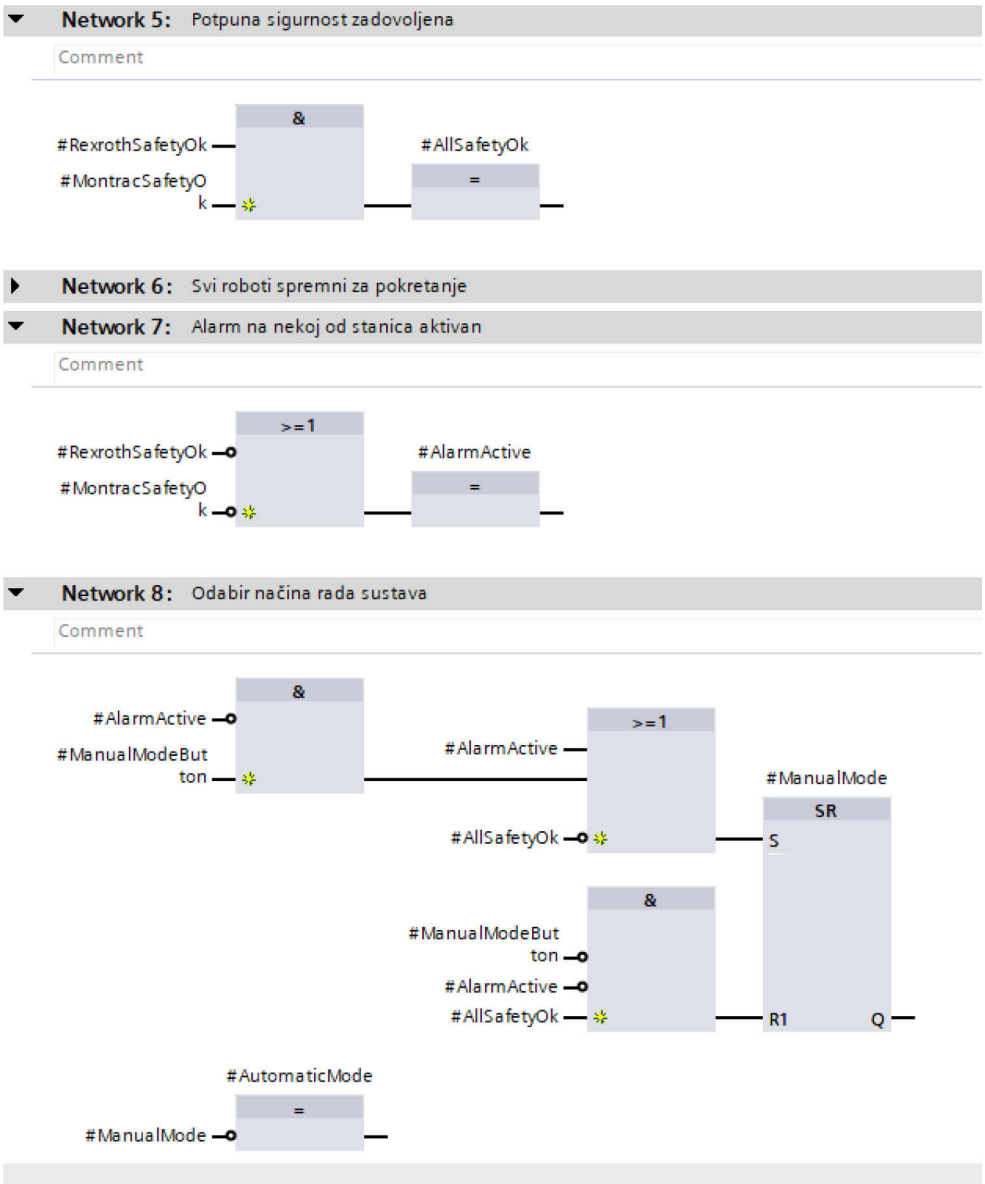
Slika 22. Funkcija za kontrolu transportnim sustavom s dodatnim ulaznim signalima unutar funkcijskog bloka stanice

3.1.1.3. Komunikacija

Logika za komunikaciju između PLC-a nije mijenjana, međutim dodani su novi podaci koji se izmjenjuju između PLC-a promjenom strukture podatkovnih blokova „PLC1 – PLC2 komunikacija“ i „PLC2 – PLC1 komunikacija“. Dodani podaci se odnose na signale neophodne za postizanje sigurnosti cijelog sustava.

3.1.1.4. Sigurnost

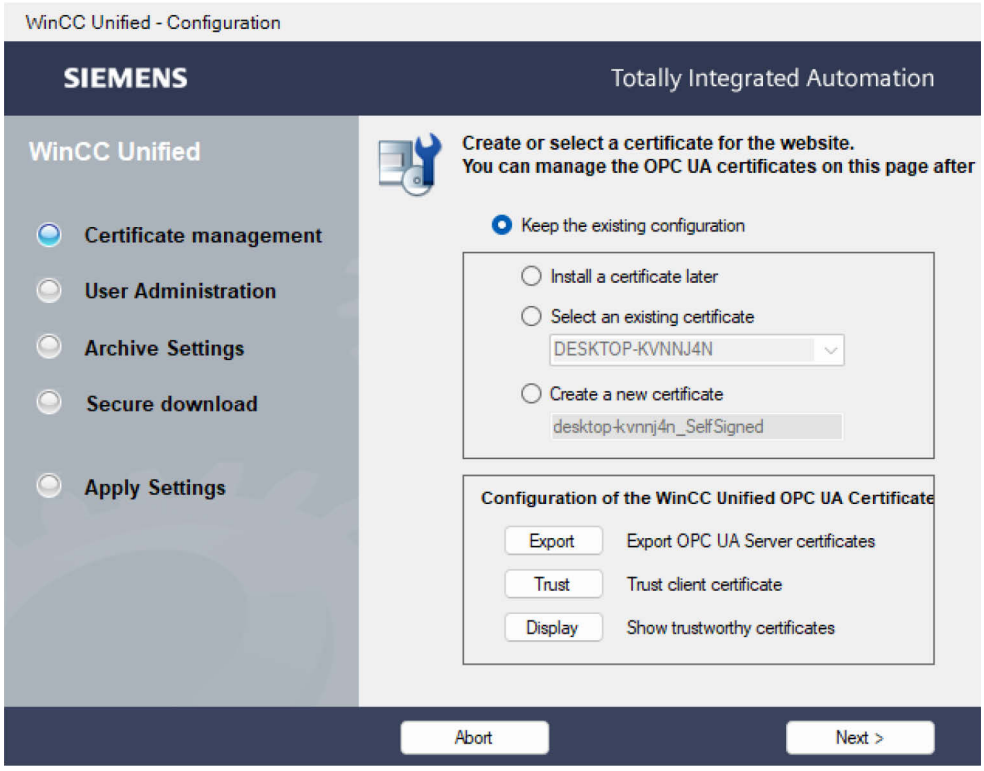
Za postizanje željene razine sigurnosti prilikom rada sustava, napravljen je funkcijski blok „SigurnostSustava“ koji kao ulazne signale uzima stanja s robota i način rada odabran u korisničkom sučelju te ih pretvara u signale koji definiraju način rada cijelog sustava. Ako je zadovoljena sigurnost na svim robotima u sustavu, funkcijski blok generira signal „AllSafetyOk“ koji omogućuje izmjenu načina rada sustava iz ručnog u automatski [Slika 23]. Ako u bilo kojem trenutku za vrijeme rada sustava u automatskom načinu navedeni signal padne u logičku nulu, rad svih robota i transportnih sustava se zaustavlja te je potrebno ukloniti razlog pada sigurnosti sustava prije nego što je moguće vratiti sustav u automatski način rada.



Slika 23. Logika za odabir načina rada ovisno o sigurnosti sustava unutar funkcijskog bloka „SigurnostSustava“

3.2. Programska podrška korisničkog sučelja

Za programiranje korisničkog sučelja korišten je Siemensov vizualizacijski alat „WinCC Unified V16“ koji je također integriran u TIA Portal. Ovo je ujedno i prva verzija WinCC Unified alata te kao takva sadrži niz programskih grešaka koje su riješene u višim verzijama. Pošto su svi TIA Portal projekti kompatibilni s višim verzijama, preporuča se da se u budućnosti projekt usavrši u nekoj od novijih verzija TIA Portal-a. Za korištenje alata u TIA Portal-u, prethodno je potrebno napraviti njegovu konfiguraciju koristeći „WinCC Unified Configuration“ aplikaciju [Slika 24].

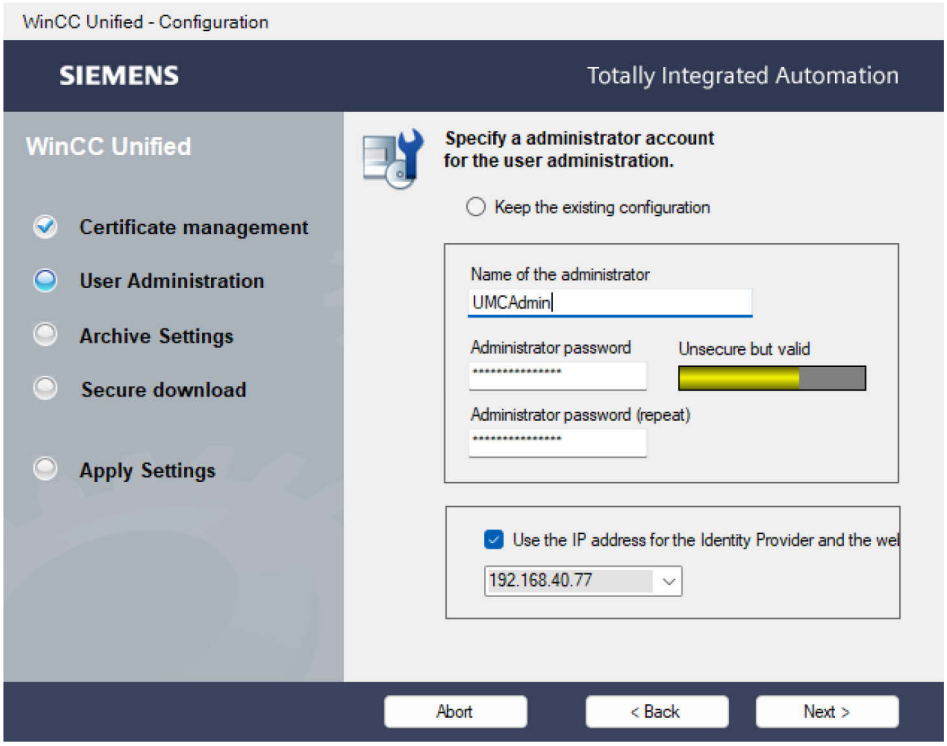


Slika 24. Sučelje „WinCC Unified Configuration“ aplikacije

Aplikacija se sastoji od četiri vrste postavki:

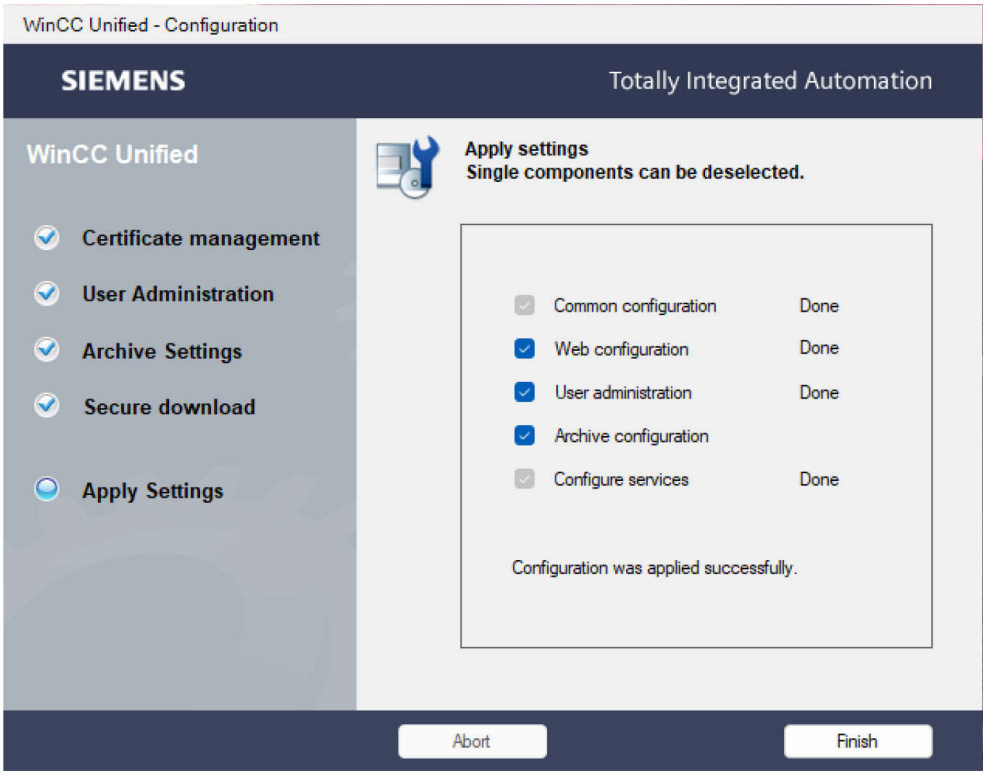
1. Upravljanje certifikatima –stvaranje ili odabir certifikata web stranice
2. Korisnička administracija – stvaranje računa administratora nužnog za administraciju korisnika HMI-a
3. Postavke arhive – odabir mjesta na računalu koje će služiti kao baza podataka i postavki SQL poslužitelja
4. Sigurno preuzimanje – odabir lozinke za sigurno preuzimanje projekta HMI-a

Postavke za upravljanje certifikatima mogu se zanemariti prilikom prve konfiguracije jer se za naknadnu instalaciju certifikata može koristiti „WinCC Unified Certificate Manager“ aplikacija. Pod postavkama korisničke administracije [Slika 25] definira se račun administratora pod imenom „UMCAdmin“ gdje UMC označava „Unified Monitoring Configuration“. Račun administratora nužan je za naknadnu administraciju korisnika HMI-a definiranih u TIA Portal projektu. Ovdje se također se nalazi postavka za odabir IP adrese web stranice sučelja. Odabrana adresa također mora biti statička adresa računala koje se koristi kao web server. U dogovoru s kolegama iz CRTA-e odabrana je IP adresa „192.168.40.77“ te će se navedena adresa kasnije koristiti kao pristup korisničkom sučelju preko bilo kojeg uređaja koje je spojeno na lokalnu mrežu CRTA prostora te koje može koristiti bilo koju vrstu web preglednika.



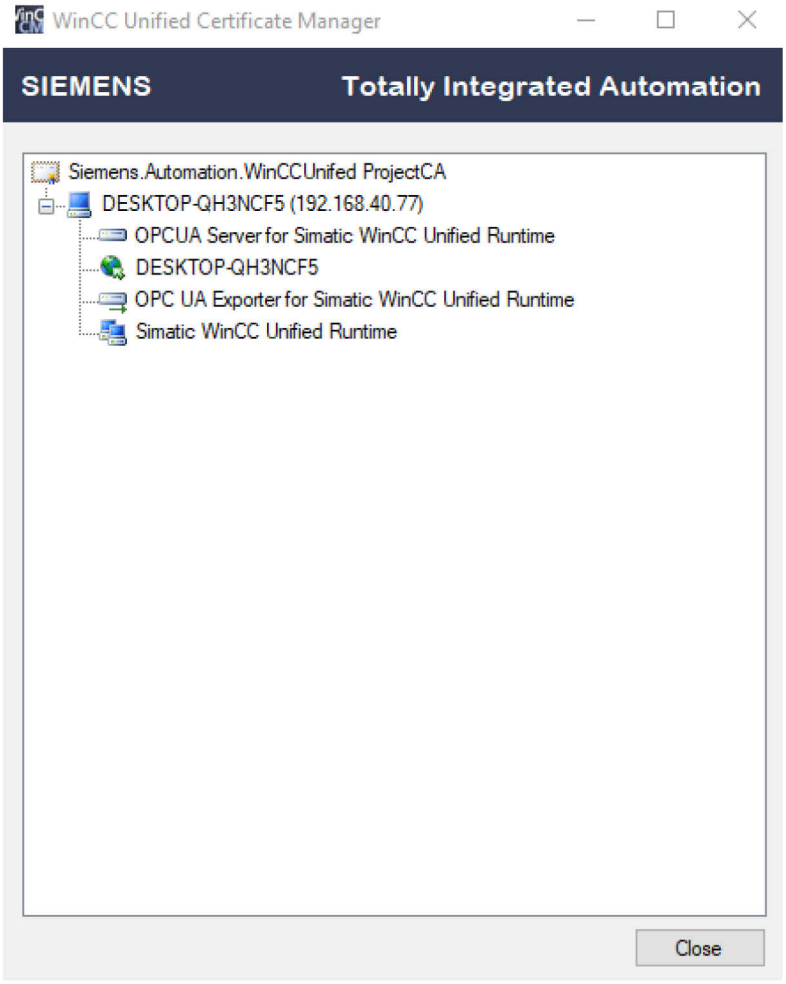
Slika 25. Postavke korisničke administracije „WinCC Unified Configuration“ aplikacije

Pod postavkama arhive potrebno je odabrati mjesto na računalu koje će služiti kao baza podataka i postavki SQL poslužitelja. Aplikacija također preporučuje korištenje tvrdog diska koje na sebi ne sadrži instalaciju Windows operativnog sustava radi optimizacije performansi. Pod postavkama sigurnog preuzimanja potrebno je definirati lozinku koja će se unositi u TIA Portal-u prilikom preuzimanja HMI projekta na web server. Nakon definiranja svih postavki, postavke je potrebno i primijeniti pritiskom tipkala „Apply“. Nakon primjene postavki, pojavit će se obavijest o uspješnoj primjeni postavki te se aplikacija može zatvoriti [Slika 26].



Slika 26. Primjena postavki „WinCC Unified Configuration“ aplikacije

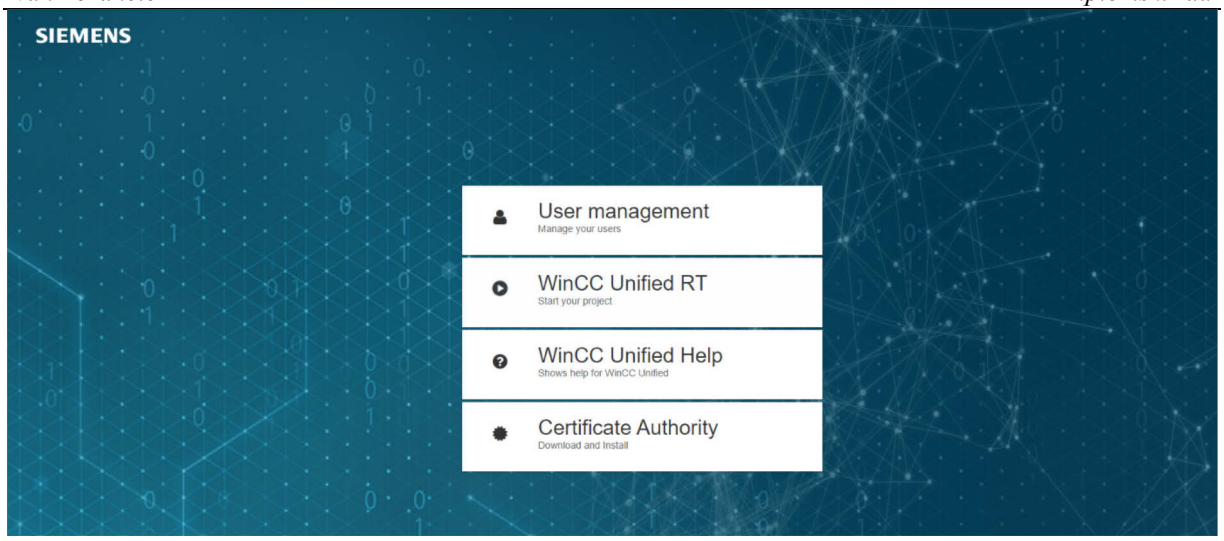
Ukoliko su preskočene postavke za upravljanje certifikatima unutar „WinCC Unified Configuration“ aplikacije, potrebno je otvoriti „WinCC Unified Certificate Manager“ te unutar aplikacije definirati i instalirati sve potrebne certifikate [Slika 27].



Slika 27. Sučelje „WinCC Unified Certificate Manager“ aplikacije

Ukoliko se nakon konfiguracije posjeti konfigurirana web stranica korisničkog sučelja, koja je u ovom slučaju https://192.168.40.77, korisnik će biti dočekan sa WinCC Unified sučeljem [Slika 28] gdje će moći:

- upravljati korisnicima preko UMCAdmin računa
- pokrenuti korisničko sučelje
- pokrenuti WinCC Unified pomoć
- instalirati certifikate za korištenje sučelja



Slika 28. Web stranica korisničkog sučelja nakon konfiguracije

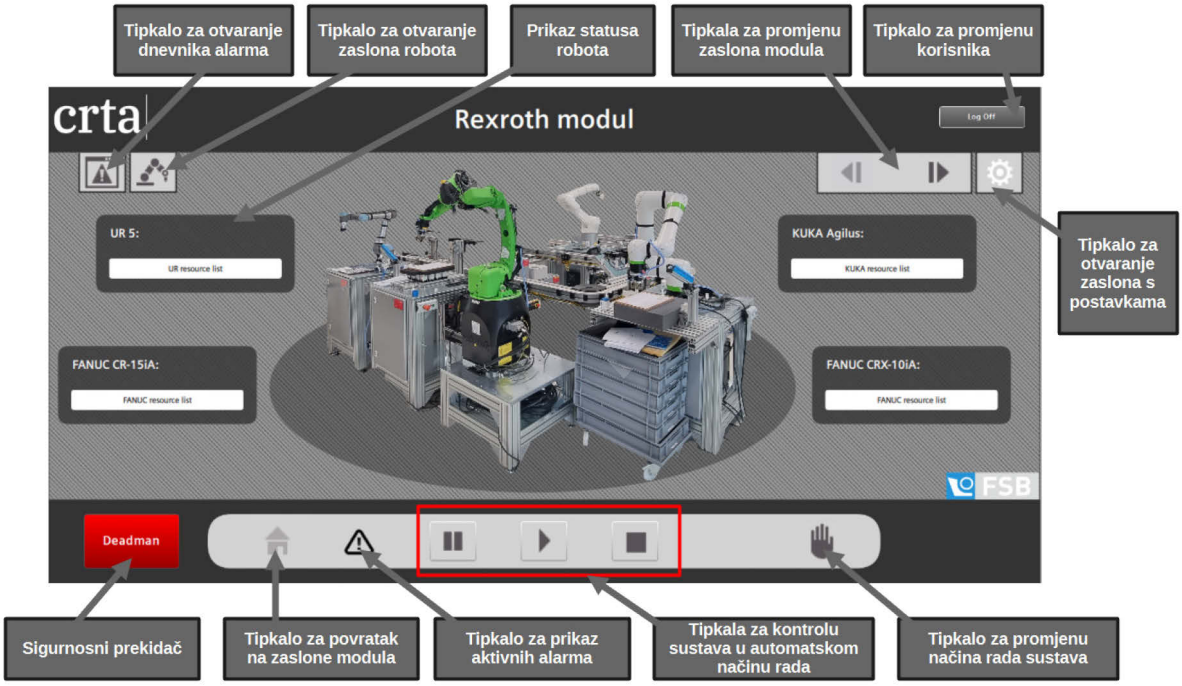
Naravno, ukoliko korisničko sučelje još nije razvijeno, pokušaj pokretanja projekta biti će neuspješan, stoga je potrebno otvoriti već definirani TIA Projekt te u njemu razviti sučelje. Za potrebe ovog diplomskog rada, zasloni korisničkog sučelja unutar TIA Portal HMI projekta organizirani su u tri grupe:

1. Zajednički zasloni
2. Montrac modul
3. Rexroth modul

gdje svaki modul ima dodatnu grupaciju u robotske i skočne (eng. *Popup*) zaslone. Unutar zajedničkih zaslona nalaze se početni zaslon, zaslon s postavkama te zaslon s dnevnikom alarma (eng. *Alarm log*). Početni zaslon [Slika 29] prvi je zaslon kojeg korisnik vidi nakon prijave u korisničko sučelje. Na početnom zaslonu nalazi se podatak o verziji sučelja koja će se svakom nadogradnjom mijenjati te samim time pratiti napredak sučelja. Na zaslonu se nalaze dva tipkala, „Log Off“ tipkalo koje će izbaciti korisnika iz trenutnog računa te „Home“ tipkalo koje otvara zaslone za module proizvodnog sustava.

Slika 29. Početni zaslon korisničkog sučelja

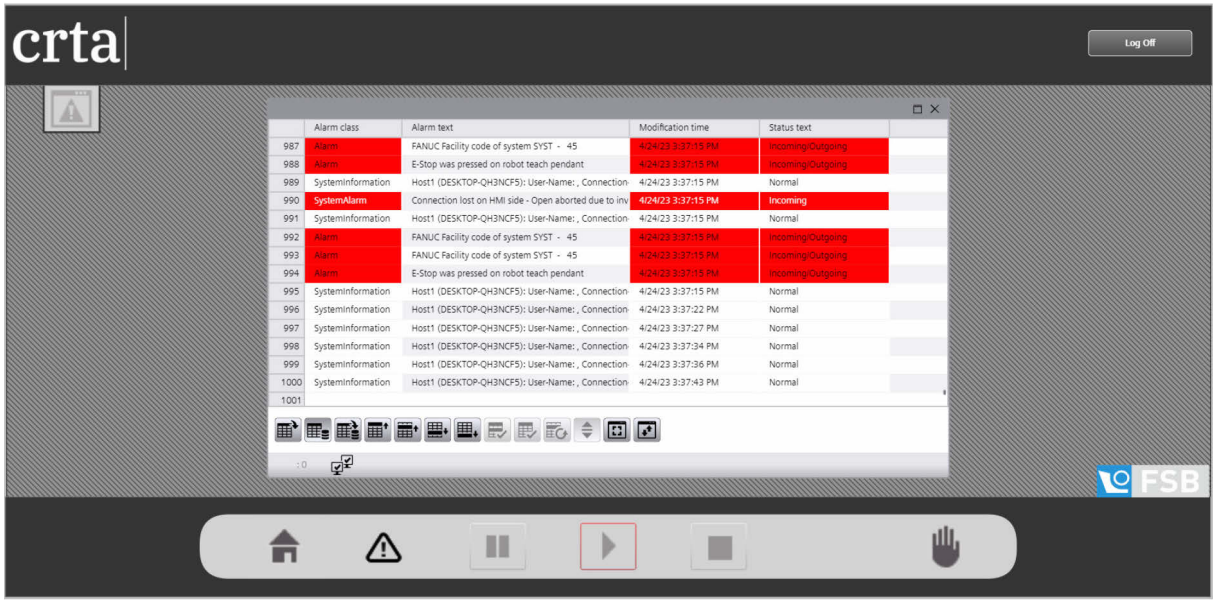
Zasloni modula proizvodnog sustava glavni su zasloni korisničkog sučelja. Na njima su prikazani stanja svih robota u sustavu te se također nalaze tipkala za pristup svim ostalim zaslonima, kao i tipkala za upravljanje i nadzor cijelog sustava. Funkcije tipkala na zaslonu modula opisane su na [Slika 30].



Slika 30. Opis funkcija tipkala na zaslonu Rexroth modula

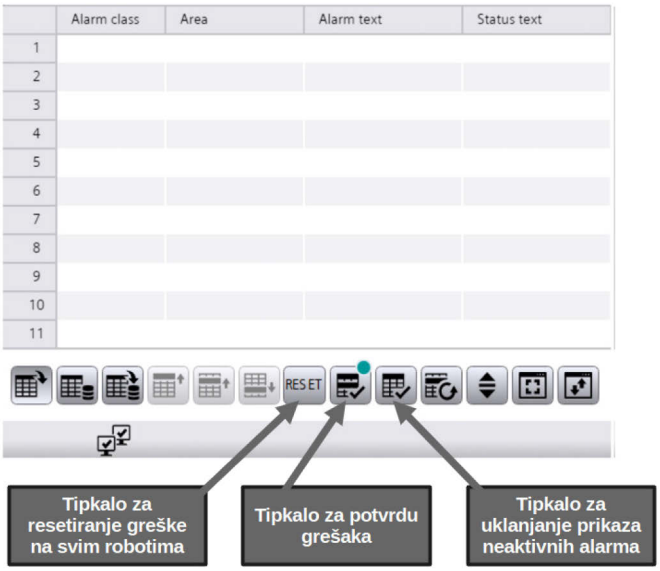
Zaslon s dnevnikom alarma [Slika 31] prikazuje povijest svih alarma kao i vrijeme njihova nastanka te njihov trenutni status. Statusi alarma mogu biti:

- Dolazni – aktiviran je alarm čiji je uzrok i dalje aktivan
- Dolazni/odlazni– aktiviran je alarm, ali je uzrok već otklonjen te se čeka potvrda alarma
- Dolazni/potvrđeni – aktiviran je alarm, uzrok je i dalje aktivan ali je alarm unaprijed potvrđen



Slika 31. Zaslon s dnevnikom alarma

Svi alarmi definirani su tako da čekaju potvrdu operatera. Operater resetira i potvrđuje alarme pomoću skočnog zaslona za aktivne alarme [Slika 32].



Slika 32. Skočni zaslon s aktivnim alarmima

Alarme s robota potrebno je definirati u obliku tekstualne liste te ih povezati s izlaznim signalom robota o aktivnoj greški koji će služiti kao okidač za prikaz signala. Tekstualne liste alarma definirane su u HMI dijelu TIA Portal projekta pod odjeljkom „Text and graphic lists“ [Slika 33], a okidač alarma se definira u odjeljku „HMI alarms“ [Slika 34]. Robot također mora slati cijeli broj, odnosno *integer*, gdje je svaki *integer* vezan za određenu grešku kako bi HMI znao koji unos s tekstualne liste je potrebno prikazati. Logika za slanje alarmnog *integera* na HMI riješena je u PLC funkcijskom bloku svakog robota.

Text lists

...	Name	Selection	Comment
	ABB_Alarm	Value/Range	
	UR_Alarm	Value/Range	
	FANUC_Alarm	Value/Range	
	KUKA_Alarm	Value/Range	
	UR_Status	Value/Range	
	ABB_Status	Value/Range	
	FANUC_Status	Value/Range	
	KUKA_Status	Value/Range	
	Mode	Value/Range	
	Uredjaj	Value/Range	

Text list entries

...	Default	Value	Text
		90	E-Stop was pressed on teach pendant
		91	Robot backup error
		92	Controller CPU fan error
		93	Program execution error
		94	Robot power fail error
		95	Production execution error
		96	Controller temperature warning
		<Add new>	

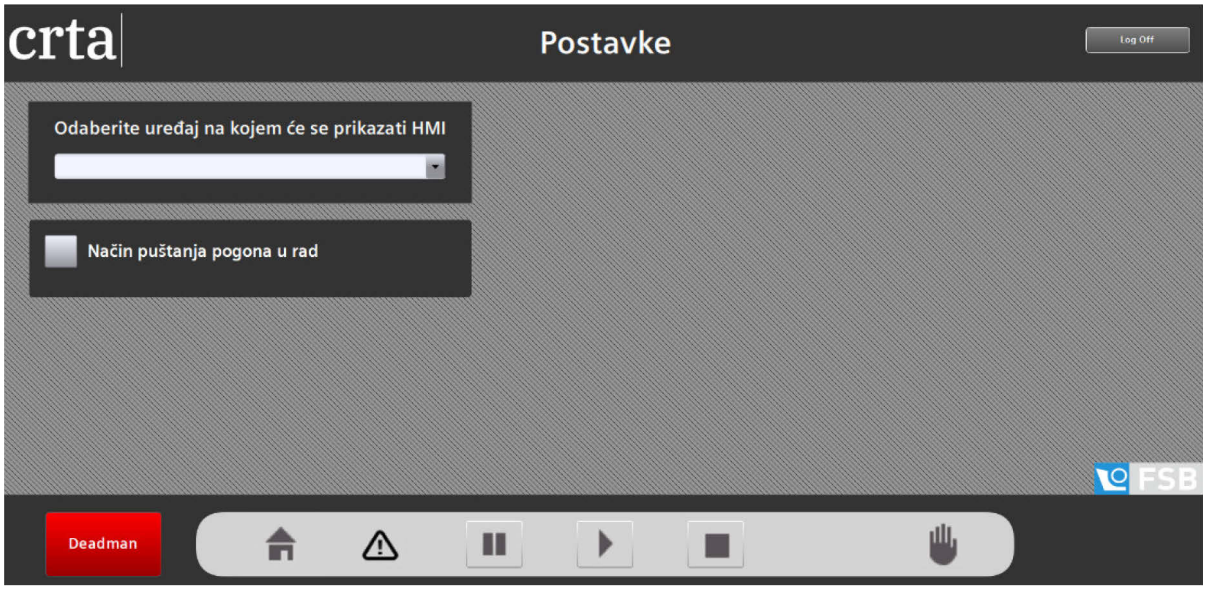
Slika 33. Definiranje alarmne tekstualne liste

Discrete alarms

ID	Name	Alarm text	Alarm class	Trigger tag	Trigger bit
15	UR5_Alarm	<Text list, UR_Alarm>	Alarm	UR5_Error	0
16	UR3_Alarm	<Text list, UR_Alarm>	Alarm	UR3_Error	0
17	CR-15iA_Alarm	<Text list, FANUC_Alarm><Tag : 5, CR-15iA_Error_code>	Alarm	CR-15iA_Error	0
18	CRX-10iA_Alarm	<Text list, FANUC_Alarm><Tag : 5, CRX10iA_Error_code>	Alarm	CRX10iA_Error	0
19	YuMi_Alarm	<Text list, ABB_Alarm>	Alarm	YuMi_Error	0
20	Agilus_Alarm	<Text list, KUKA_Alarm>	Alarm	Agilus_Error	0
<Add new>					

Slika 34. Definiranje okidača alarma

Zaslon s postavkama rada sustava [Slika 35] sadrži postavku vezanu za odabir uređaja na kojem se prikazuje korisničko sučelje te postavku vezanu za način puštanja pogona u rad (eng. *Commissioning mode*). Ovisno o odabranom uređaju za prikaz korisničkog sučelja, mijenja se sigurnosna logika za upravljanje sustavom. Naime, ukoliko se odabere uređaj koji posjeduje ekran na dodir, za upravljanje sustavom mora se jednom rukom pridržavati sigurnosni prekidač (eng. *Dead man's switch*), a drugom rukom izvršavati željene funkcije. Međutim, WinCC Unified V16 verzija ne podržava *multi-touch* opciju te se preporuča zasad ne mijenjati navedenu opciju, odnosno koristiti opciju „Laptop“ iako se koristi uređaj na dodir, sve dok se ne izvrši nadogradnja WinCC Unified alata na višu verziju. Odabirom opcije „Laptop“ sigurnosni prekidač ignorira se te nestaje iz prikaza korisničkog sučelja. Postavka vezana za način puštanja pogona u rad omogućuje testiranje rada svakog robota pojedinačno pomoću HMI ploče (eng. *faceplate*) za testiranje sekvence koja se pojavi na svakom robotskom zaslonu ukoliko je opisana postavka uključena.

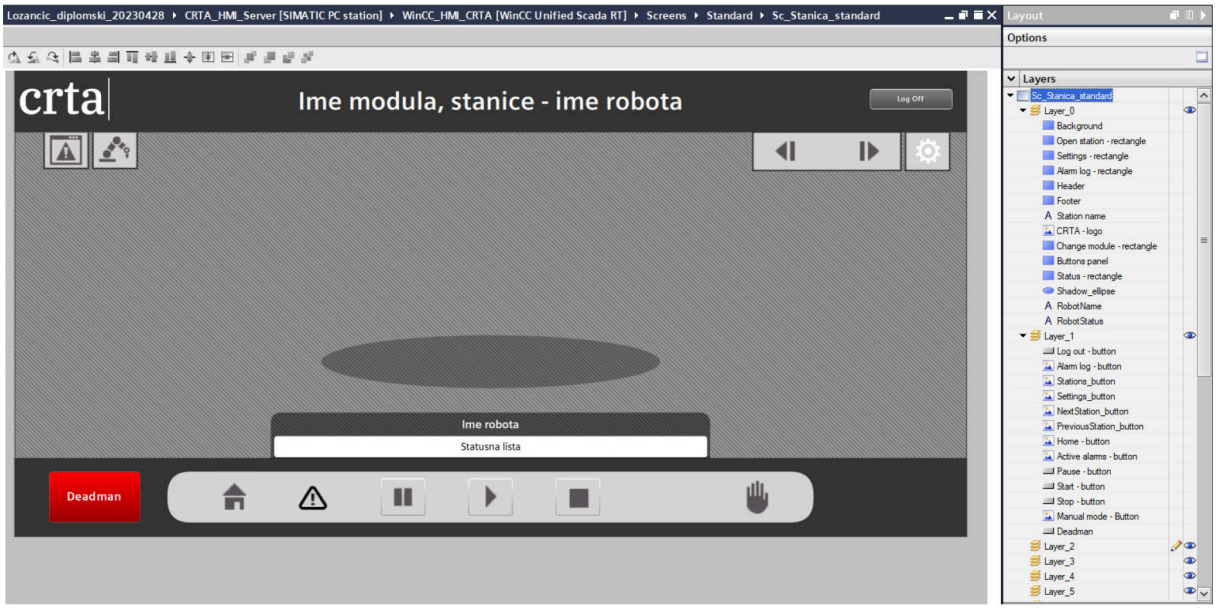


Slika 35. Zaslon s postavkama rada sustava

Kao što je već spomenuto, projekt je rađen imajući na umu postupak standardizacije PLC i HMI projekta. U tu svrhu, zasloni za robote te njihovi skočni zasloni su standardizirani. Prilikom izrade zaslona za robote obraćena je pozornost na raspored (eng. *layout*) svih HMI objekata [Slika 36]. Objekti su raspoređeni u dva sloja:

1. Sloj 0 – sadrži objekte za osnovnu vizualizaciju poput pozadine, CRTA logoa, natpisa na zaslonu i slično
2. Sloj 1 – sadrži sve interaktivne objekte, odnosno tipkala svakog HMI zaslona

Svi dodatni objekti potrebni za upravljanje i nadzor robota naknadno se dodaju u druge slojeve po izboru.

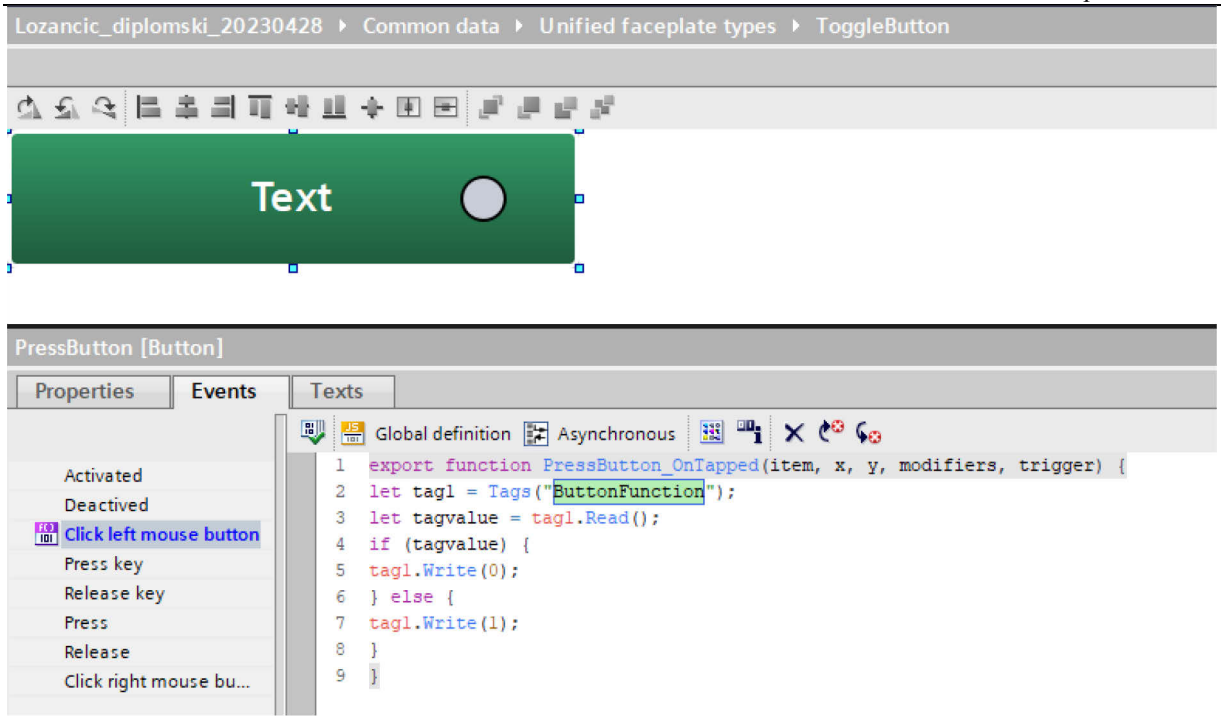


Slika 36. Standardizirani zaslon za robote

Nužni objekti su također standardizirani u obliku *faceplate*-ova. *Faceplate* označava skup osnovnih HMI elemenata koji objedinjuje funkcije elemenata u jednu. Napravljeno je pet različitih *faceplate*-ova:

1. Tipkalo s prikazom stanja senzora
2. Tipkalo bez prikaza stanja
3. Prekidač s prikazom stanja senzora
4. Prekidač za kontrolu hvataljke bez grafičkog prikaza
5. Tester sekvence

Sva logika vezana za *faceplate*-ove napisana je u JavaScript-u. Logika je često vrlo jednostavna, primjerice logika za prekidač s prikazom stanja senzora funkcionira tako što čita stanje definiranog signala te pritiskom na gumb mijenja njegovo logičko stanje iz logičke jedinice u logičku nulu i obrnuto [Slika 37].

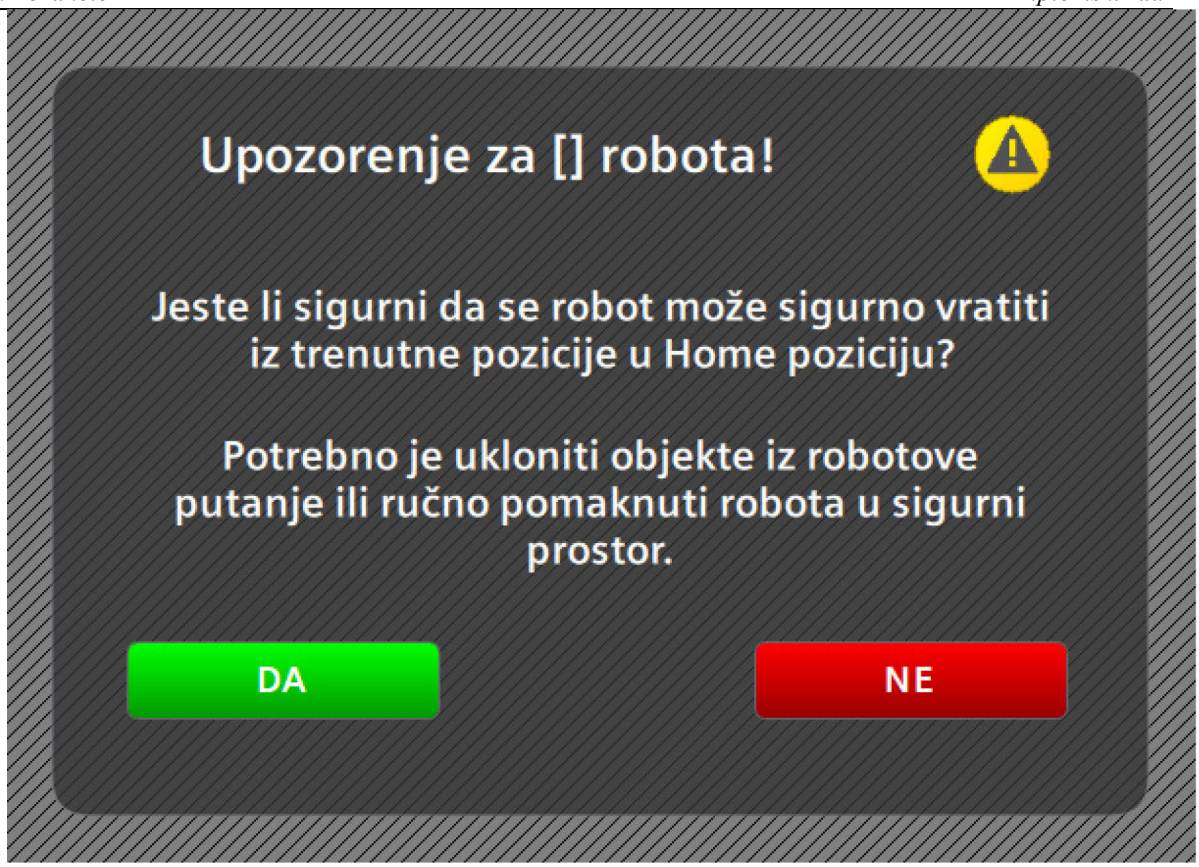


Slika 37. Logika za definiranje prekidača s prikazom stanja

Definirani *faceplate*-ovi najčešće koriste četiri ulazna signala:

1. LED – grafički prikaz stanja sa senzora
2. ButtonFunction – signal čija se logička vrijednost mijenja pritiskom tipkala
3. ButtonText – tekst prikazan na tipkalu
4. AllowControl – signal koji omogućuje korištenje *faceplate*-a ovisno o načinu rada

Faceplate se koristi tako što se ubaci na željeni zaslon te mu je samo potrebno definirati prethodno objašnjene ulazne signale. Ovime se uvelike smanjuje proces stvaranja novih funkcija upravljanja sustavom te se znatno povećava produktivnost programera sučelja. Osim zaslona i *faceplate*-ova, standardiziran je i skočni zaslon za upozorenje vezano za rad robota [Slika 38]. Definirani skočni prozori služe kao dodatna razina zaštite od slučajnog pomicanja robota prilikom korištenja sučelja.



Slika 38. Skočni prozor za upozorenje pri pomicanju robota u početni položaj

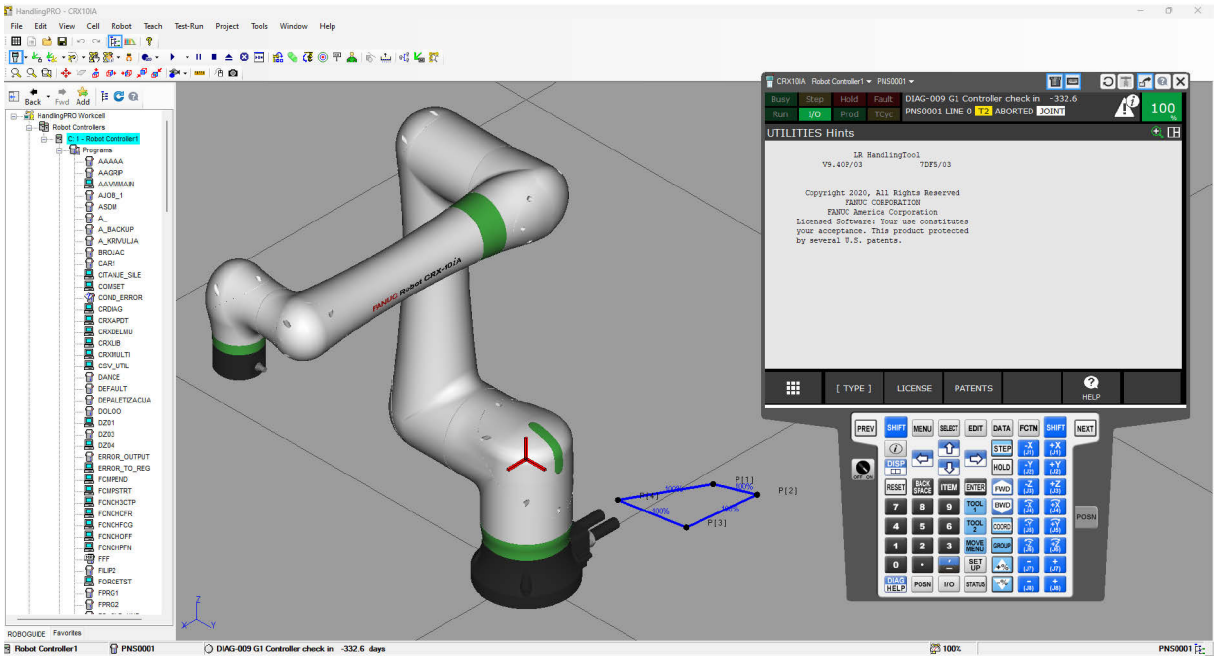
3.3. Programska podrška robota

3.3.1. Fanuc kolaborativni roboti

Zbog identičnog načina rada Fanuc kolaborativnih robota CR i CRX serije, programska podrška za Fanuc robote biti će u potpunosti opisana samo na primjeru CRX-10iA robota, imajući na umu da je potpuno identičan postupak konfiguracije i programiranja obavljen i na CR-15iA robotu.

3.3.1.1. Korištenje ROBOGUIDE programskog alata

U svrhu simulacije i programiranja Fanuc robota, korišten je službeni programski alat tvrtke Fanuc naziva ROBOGUIDE. Stvaranjem sigurnosne kopije fizičkog robota moguće je napraviti projekt unutar ROBOGUIDE alata koji će simulirati virtualnu kopiju robota [Slika 39].

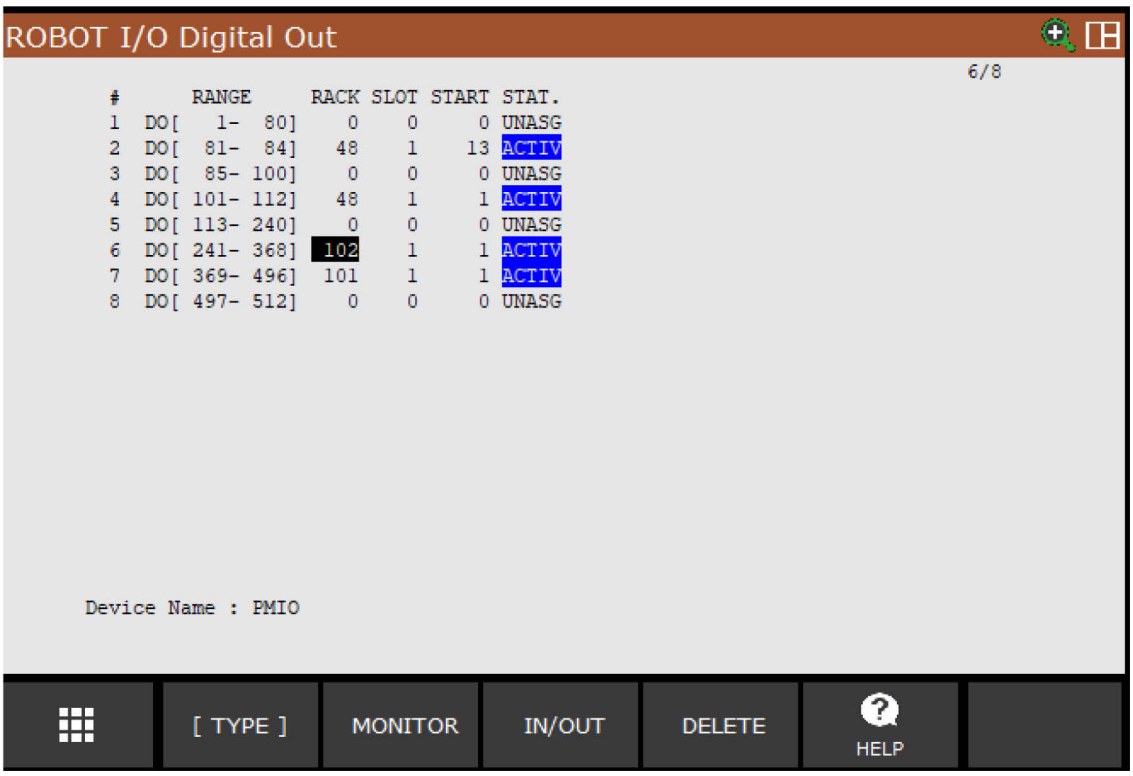


Slika 39. Simulacija CRX-10iA robota pomoću ROBOGUIDE programskog alata

Program je prvenstveno korišten za testiranje programa i promjena postavki koje bi nepravilnom uporabom mogle negativno utjecati na rad robota. Također, za programiranje specifičnih funkcija Fanuc robota nužno je koristiti programski jezik KAREL, a za prevođenje KAREL programa u programe razumljive robotu koristio se upravo ROBOGUIDE.

3.3.1.2. Programiranje na strani robota

Prije programiranja, potrebno je konfigurirati komunikaciju između robota i PLC-a. Željeni signali koji se šalju PLC-u definirani su kao digitalni izlazi na stalku (eng. *rack*) 102 koji u Fanuc robotima označuje I/O uređaj (eng. *I/O Device*) [Slika 40].



Slika 40. Konfiguracija digitalnih izlaza Fanuc robota za komunikaciju s PLC-om

Kod CRX-10iA robota digitalni izlazi u rasponu 241-368, odnosno od 128 bitova, šalju se preko PROFINET-a konstantno kao digitalni ulazi PLC-u. To znači da će 241. digitalni izlaz robota imati adresu I136.0 na PLC-u, odnosno prvi definirani bit robota [Slika 41].

	Module	Rack	Slot	I address	Q address	Type	Article no.
	Fanuc CRX-10iA	0	0			A05B-2600-R834: ...	A05B-2600-R834
	FANUC Robot Controller	0	0 I/F			r30ib-plus-iodevice	
	16 Output bytes_1	0	1		126...141	16 Output bytes	A05B-2600-R834
	16 Input bytes_1	0	2	136...151		16 Input bytes	A05B-2600-R834

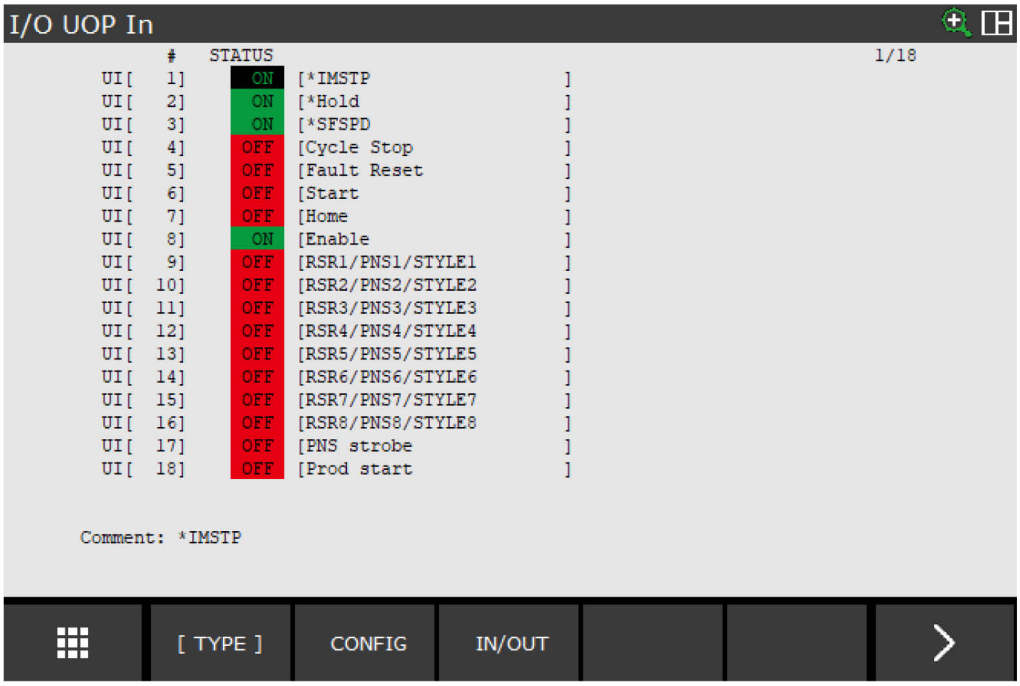
Slika 41. Konfiguracija digitalnih izlaza i ulaza Fanuc robota na strani PLC-a

Za kontrolu robota preko PLC-a potrebno je promijeniti nekoliko postavki na robotu. Koristeći „System/Config“ zaslon potrebno je:

- Opciju „Enable UI signals“ staviti u „TRUE“
- Opciju „START for CONTINUE only“ staviti u „TRUE“
- Opciju „CSTOPI for ABORT“ staviti u „TRUE“

- Opciju „Abort all programs by CSTOPI“ staviti u „FALSE“
- Opciju „Remote/Local setup“ staviti u „Remote“
- Opciju „UOP auto assignment“ staviti u „Full“

Kratica UOP označava korisničku upravljačku ploču (eng. *User Operator Panel*) te sadrži sve signale potrebne za daljinsku kontrolu robota preko PLC-a [Slika 42]. Nakon aktivacije opisanih signala na „System/Config“ zaslonu, signale je potrebno i mapirati koristeći „I/O UOP Config“ zaslon gdje je UOP signale potrebno mapirati na neke od prethodno konfiguriranih digitalnih signala.

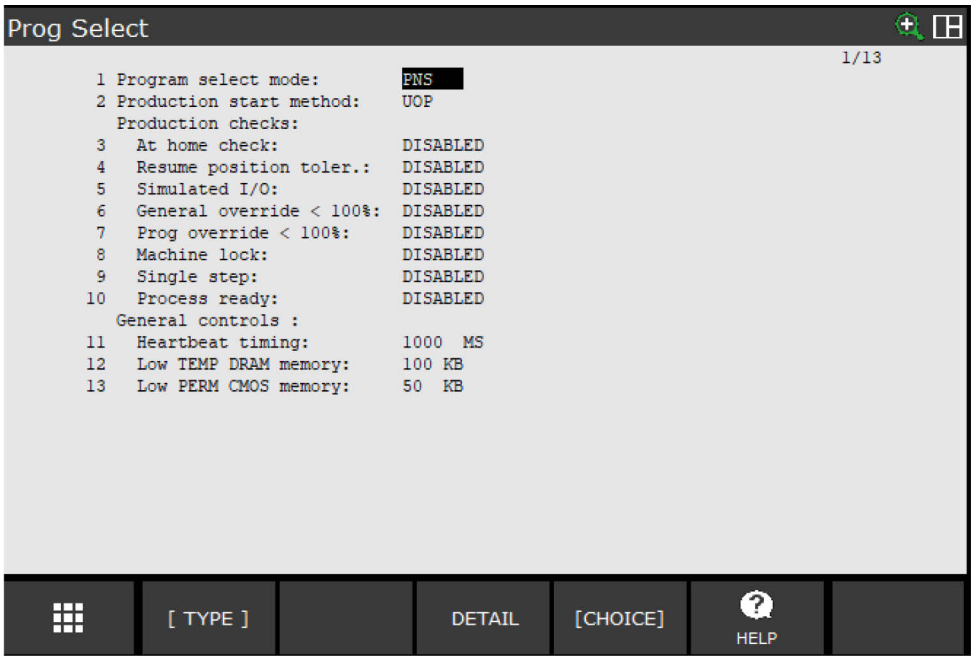


Slika 42. UOP ulazni signali za kontrolu Fanuc robota preko PLC-a

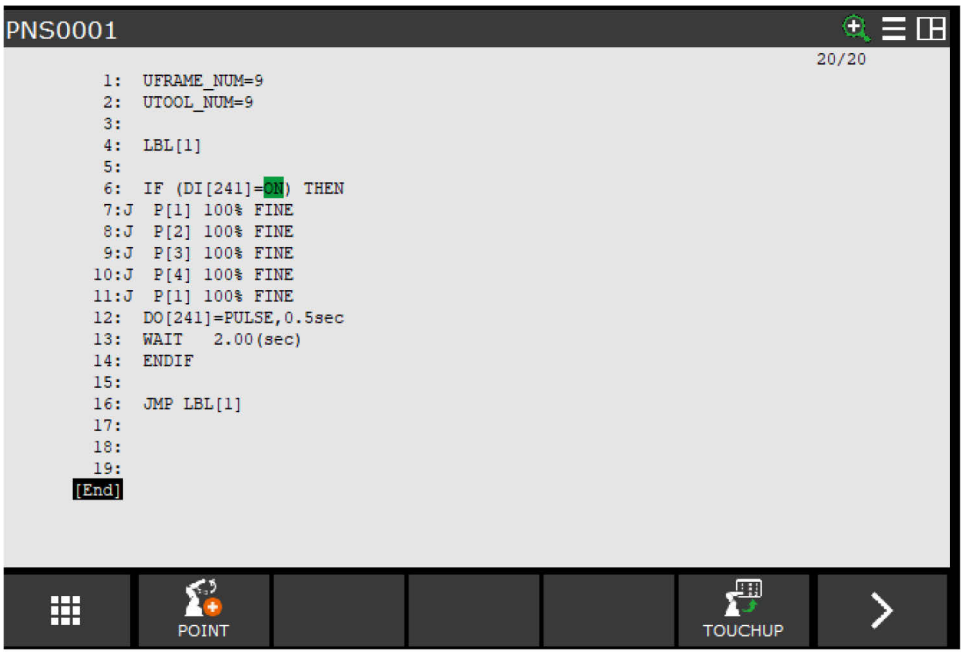
Kontrola Fanuc robota preko PLC-a može se ostvariti koristeći jednu od sljedećih metoda:

- RSR (eng. *Robot Service Request*) – omogućuje definiranje 8 numerički identificiranih programa koji odgovaraju 8 ulaznih bitova (UI signali od 9 do 16)
- PNS (eng. *Program Number Select*) –slična RSR metodi, međutim 8 bitova se identificira kao binarni broj
- STYLE – slična PNS metodi, ali se ne koristi nazivlje za odabir programa, već je potrebno definirani tablicu koja pridružuje program s brojem željenog UOP programa
- OTHER – najjednostavnija metoda koja pokreće eksplicitno navedeni program nakon što robot dobije signal za pokretanje

Robot je konfiguriran tako da koristi PNS metodu pokretanja koja se prikazala kao najbolji omjer jednostavnosti i fleksibilnosti. Konfiguracija se postiže promjenom postavki na „Prog Select“ zaslonu [Slika 43] te stvaranjem PNS0001 programa koji će služiti kao glavni robotski program. Glavni program je vrlo jednostavan, prilikom izvršavanja programa robot čeka ulazni signal koji označuje dolazak palete u radni položaj na transportnom sustavu, nakon čega robot obavlja jednostavnu kretnju te potom daje izlazni signal za otključavanje palete i omogućuje njeno kretanje na sljedeću stanicu [Slika 44].

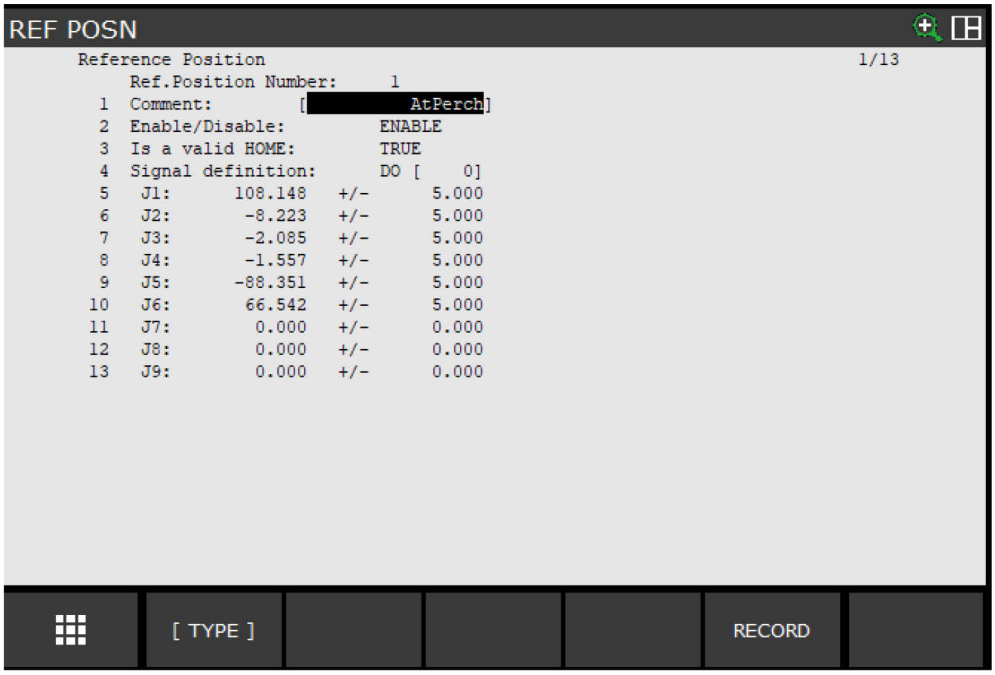


Slika 43. Konfiguracija postavki na „Prog Select“ zaslonu Fanuc robota

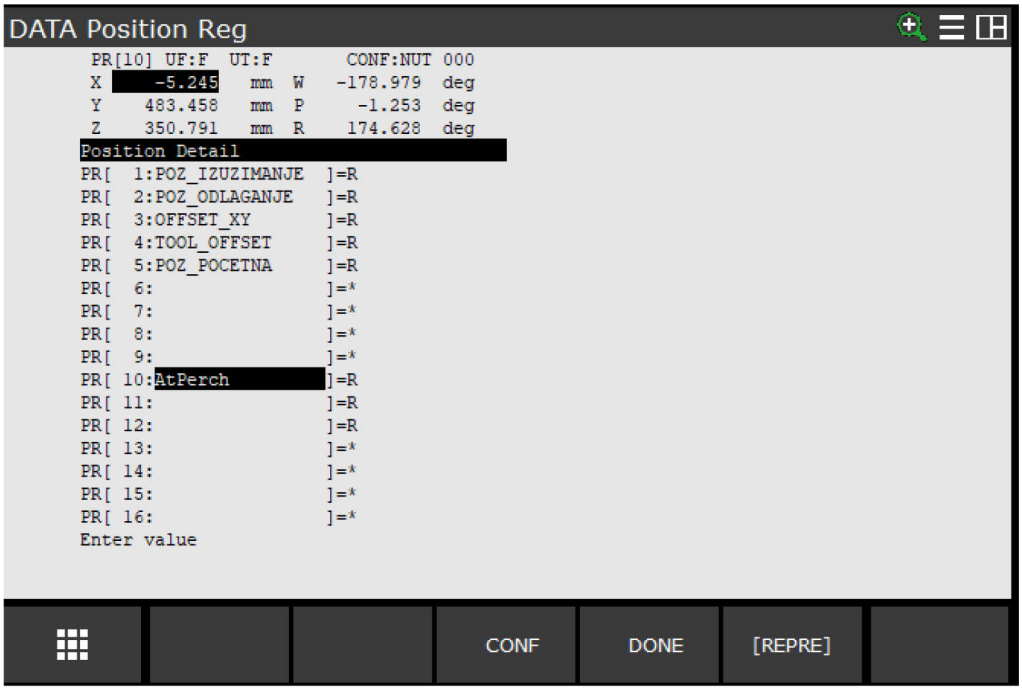


Slika 44. Definirani PNS0001 program Fanuc robota

Osim glavnog programa, potrebno je imati program za vraćanje robota u početni položaj, odnosno *homing* program. Prije pisanja programa potrebno je stvoriti referentni položaj koji će biti definiran kao početni položaj robota [Slika 45]. Za pozivanje definiranog položaja iz robotskog programa potrebno je također korištenjem „DATA Position Reg“ zaslona spremiti početni položaj robota u registar položaja [Slika 46].

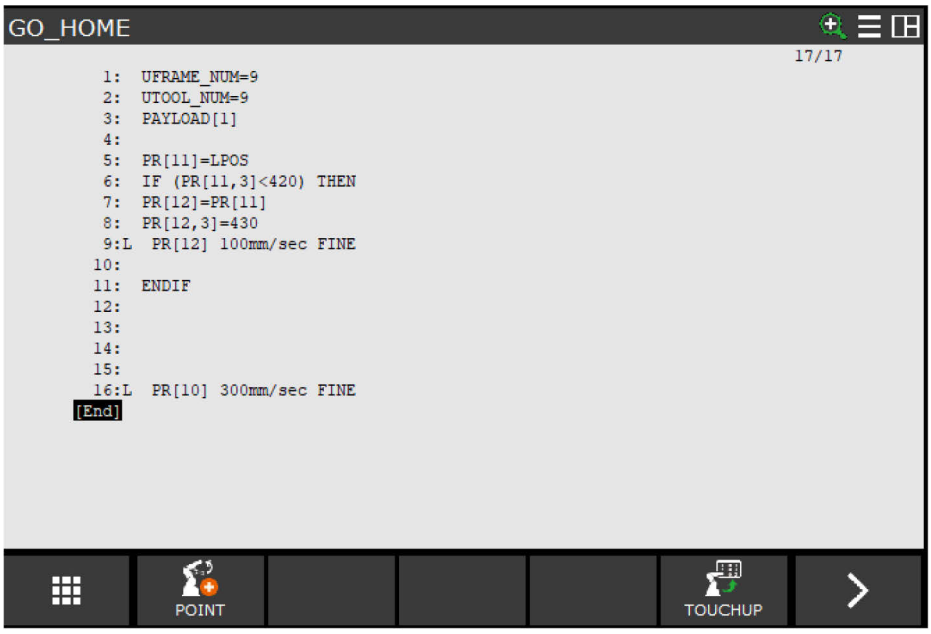


Slika 45. Definiranje početnog položaja Fanuc robota



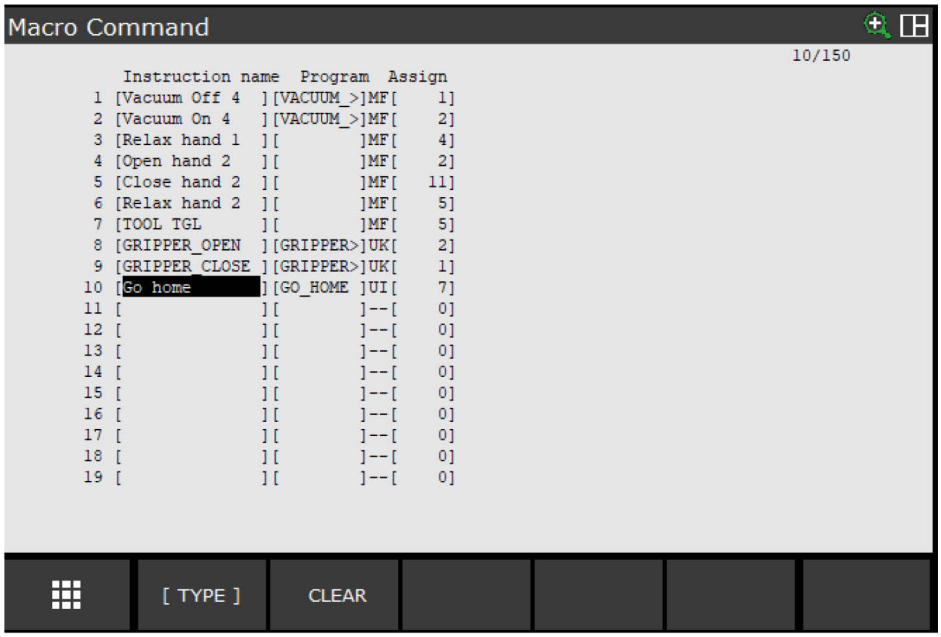
Slika 46. Spremanje početnog položaja Fanuc robota u registar položaja

Homing program napravljen je imajući na umu nemogućnost direktnog pozivanja robota u početni položaj zbog opasnosti od kolizije. Zbog toga *homing* program provjerava položaj robota po Z osi te ukoliko je njegov položaj ispod neke granične vrijednosti, prije povratka u početni položaj robot se podiže po Z osi iznad granične vrijednosti [Slika 47].



Slika 47. Definiranje *homing* programa za Fanuc robote

Konačno je *homing* program potrebno povezati i s UOP ulaznim UI[7] (*Home*) signalom pomoću makro naredbe [Slika 48]. UI[7] signal će u svom visokom stanju slati robota u početni položaj. Kada robot zauzme početni položaj, automatski će se upaliti UO[7] (*At Perch*) izlazni signal koji se šalje PLC-u.



Slika 48. Definiranje makro naredbe *homing* programa Fanuc robota

Pošto je u cilju slati sve robotske greške na PLC radi prikaza greški na HMI-u, potrebno je napraviti i konfiguraciju koja će omogućiti slanje oznake pogreške s robota na PLC. Greške na Fanuc robotima označuju se pomoću dva koda, gdje jedan označuje kod objekta (eng. *facility code*), a drugi kod pogreške (eng. *error code*). Za očitavanje navedenih kodova potrebno je osmisliti program koji će kodove zapisivati u registar iz kojih će se podaci preslikavati na grupne izlaze te kao takvi čitati na strani PLC-a. U tu svrhu se koristi ERROR_TO_REG KAREL program koji očitava sistemske oznake aktivne pogreške te omogućava njihovo zapisivanje u registar. Potpuni ERROR_TO_REG program nalazi se u prilogu, a program je prije korištenja prvo potrebno prevesti pomoću ROBOGUIDE-a u PC program razumljiv robotu. Također je potrebno definirati dva dodatna programa:

1. Program praćenja stanja (eng. *condition monitor program*) COND_ERROR [Slika 49]
2. Program stanja (eng. *condition program*) ERROR_OUTPUT [Slika 50]

COND_ERROR program pozivat će program stanja ERROR_OUTPUT svaki put kada dođe do pogreške na robotu nadzirući promjenu izlaznog signala UO[6] (*Fault*). ERROR_OUTPUT zatim će pozivati prethodno definirani ERROR_TO_REG program te preslikavati vrijednosti iz registara u grupne izlazne signale.

```
COND_ERROR

   1:  WHEN UO[6]=ON+,CALL ERROR_OUTPUT
   2:  WHEN UO[6]=OFF-,CALL ERROR_OUTPUT
[End]
```

Slika 49. Definiranje COND_ERROR programa Fanuc robota

```
ERROR_OUTPUT

   1:  CALL ERROR_TO_REG(20)
   2:  GO[1]=R[20]
   3:  GO[2]=R[21]
   4:  MONITOR COND_ERROR
[End]
```

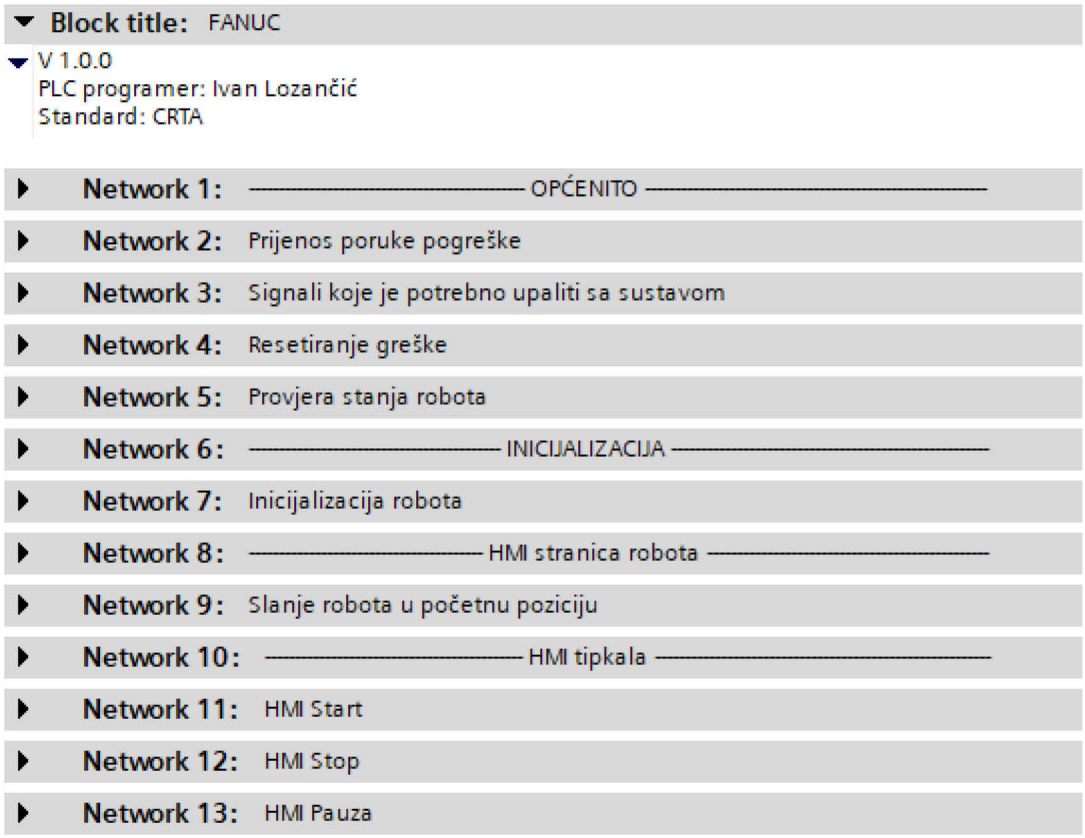
Slika 50. Definiranje ERROR_OUTPUT programa Fanuc robota

Naposlijetku preostaje aktivirati COND_ERROR program pomoću „System monitor“ zaslona. Ukoliko je opcija aktivacije programa neomogućena na robotu, potrebno je pokrenuti robota „kontroliranim startom“ (eng. *controlled start*) te mu promijeniti iznos sistemske varijable $TPP_MON.$GLOBAL_MT u 2. Nakon promjene, robot se može ponovno pokrenuti „hladnim startom“ (eng. *cold start*) te je potrebno ponoviti postupak aktivacije programa praćenja stanja.

3.3.1.3. Programiranje na strani PLC-a

Na strani PLC-a napravljen je standardizirani funkcijski blok koji koristi PNS metodu za pokretanje glavnog programa na robotu. Funkcijski blok kao ulazne signale prima prethodno definirane UO signale na robotu te šalje UI signale za daljinsko upravljanje robotom. Svaki standardizirani funkcijski blok robota podijeljen je u četiri odjeljka [Slika 51]:

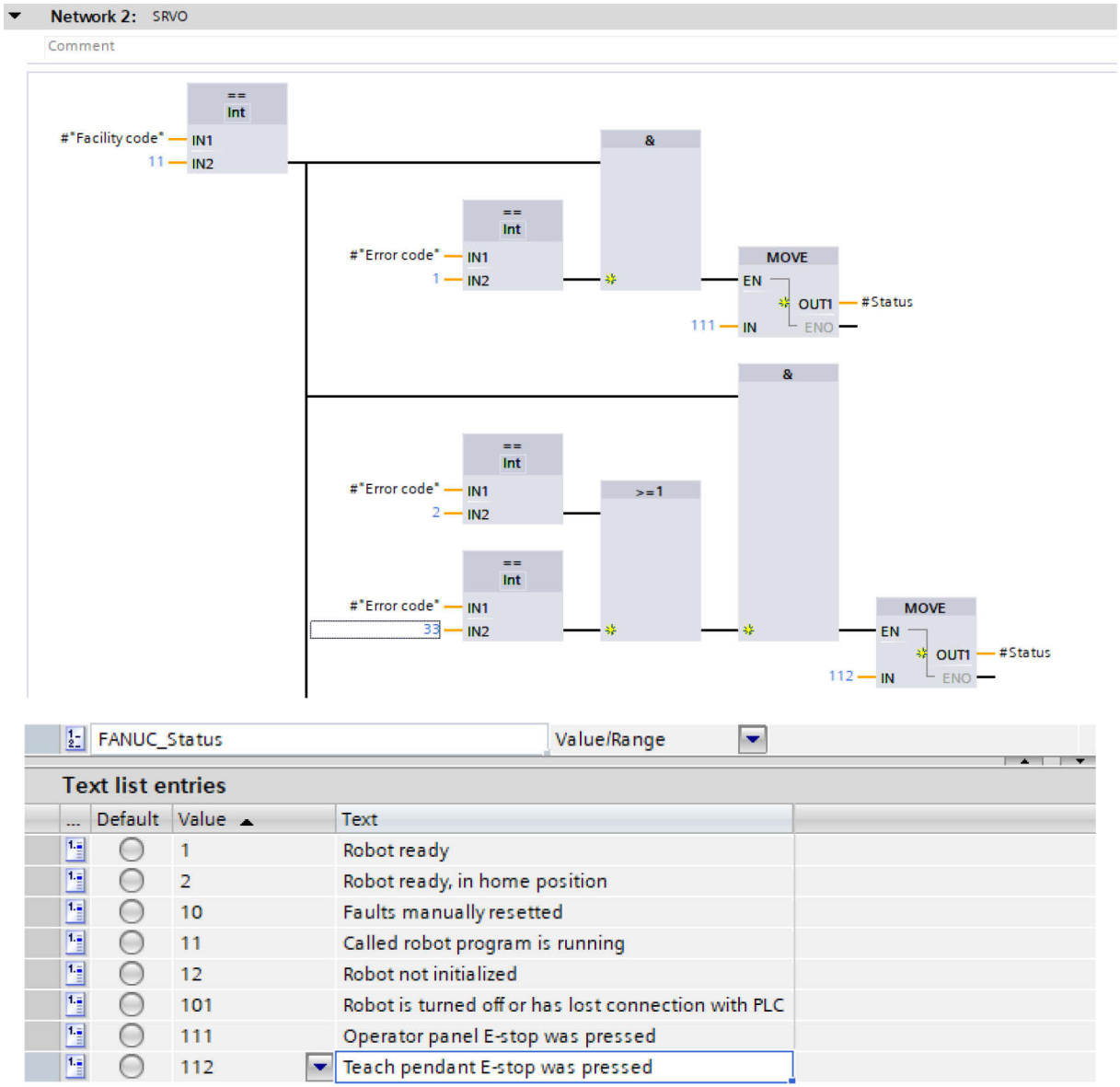
1. Općenito – logika vezana za ispisivanje pogreške na HMI-u, kao i resetiranje greške i provjere stanja robota
2. Inicijalizacija – logika za inicijalizaciju robota
3. HMI stranica robota – logika vezana za slanje robota u početni položaj te u nekim slučajevima dodatna logika koja se može aktivirati s HMI zaslona robota
4. HMI tipkala – logika za pokretanje, pauziranje i potpuno zaustavljanje rada robota u automatskom načinu rada



Slika 51. Funkcijski blok za upravljanje i nadzor Fanuc robota

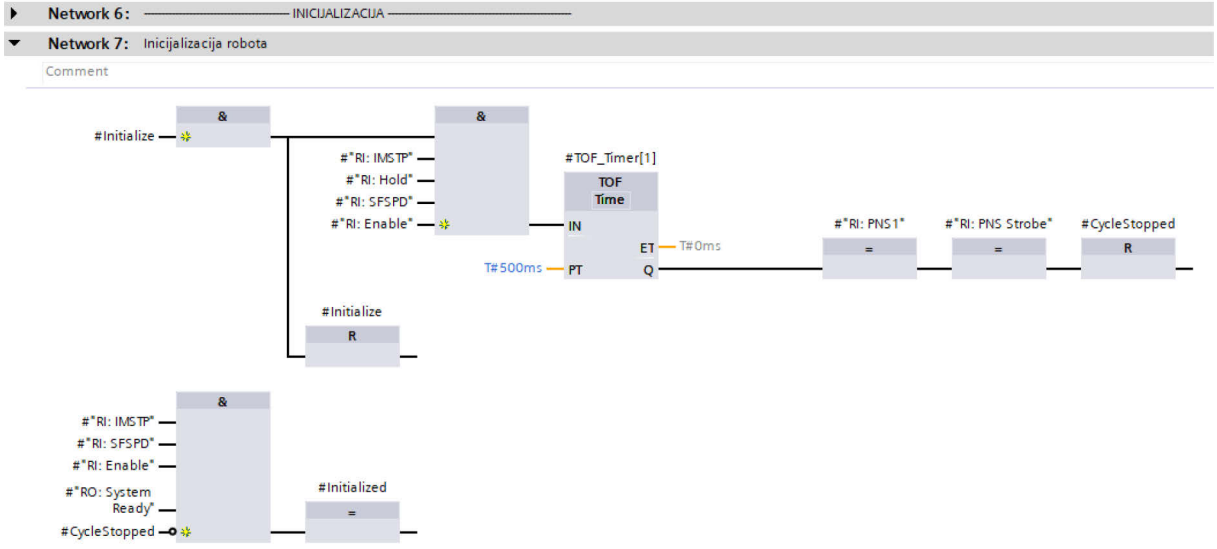
Kod prijenosa poruke pogreške, nalazi se dodatni funkcijski blok „Fanuc_msg“ koji se brine o ispisivanju trenutnog statusa robota. Logika je jednostavna, funkcijski blok na ulaze dobiva kod objekta, kod pogreške i dodatne druge signale vezane za stanje robota te s obzirom na njihove

odnose mijenja vrijednost statusnog *integera* povezanog na tekstualnu listu koja je definirana na strani HMI-a. Primjerice, ukoliko robot šalje kod objekta:11 i kod greške:2 ili 33, funkcijski blok u statusni *integer* prepisuje vrijednost 112 koja je u tekstualnoj listi definirana kao statusna poruka da je pritisnut E-Stop tipkalo na upravljačkoj jedinici (eng. *teach pendant*) robota [Slika 52].



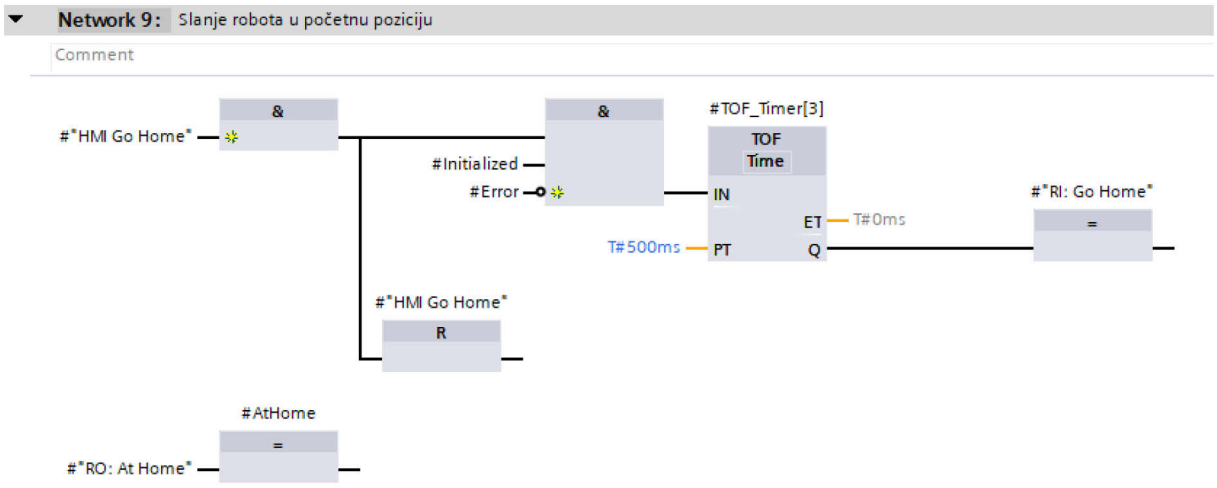
Slika 52. Logika za ispisivanje statusa Fanuc robota na HMI-u

Kako bi robot mogao biti upravljan u automatskom načinu rada sustava, prethodno se robot mora inicijalizirati i vratiti u početni položaj. Pritiskom gumba „Incijalizacija“ na HMI zaslonu robota pokreće se logika koji odabire PNS0001 program kao glavni program za izvršavanje te priprema robota za automatski način rada. Funkcijski blok također posjeduje izlaznu varijablu „Initialized“ kako bi se grafički prikazala uspješna inicijalizacija na HMI-u [Slika 53].



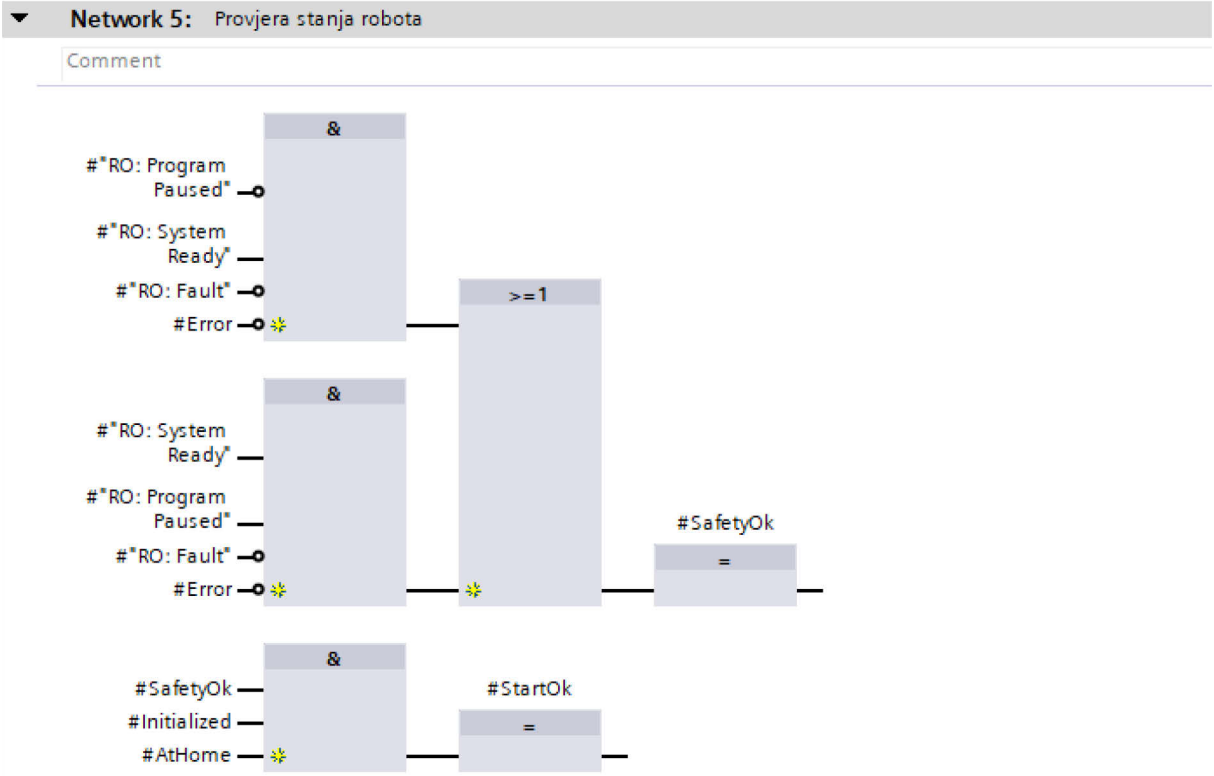
Slika 53. Logika za inicijalizaciju Fanuc robota

Pritiskom gumba „Početna pozicija“ na HMI zaslonu robota, robot dobije prethodno definirani UI[7] signal te se pokreće program za postavljanje robota u početni položaj (eng. *homing*).



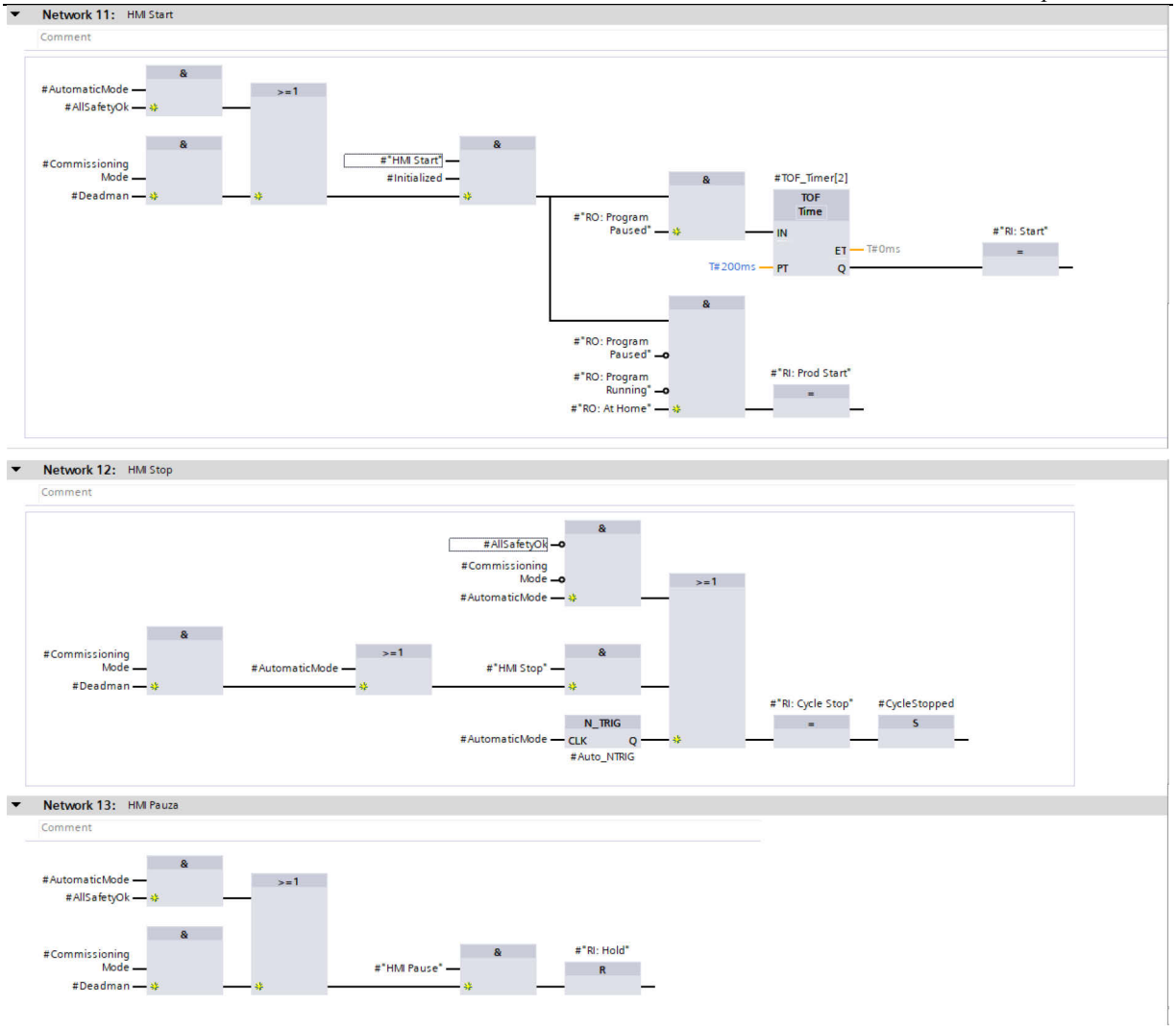
Slika 54. Logika za slanje Fanuc robota u početni položaj

Ukoliko je robot inicijaliziran i nalazi se u početnom položaju te mu je zadovoljena sigurnost, funkcijski blok aktivira izlazne signale „SafetyOk“ i „StartOk“ [Slika 55] koji se koriste unutar prethodno opisanog funkcijskog bloka „SigurnostSustava“ za omogućavanje pokretanja sustava u automatskom načinu rada. Ukoliko u bilo kojem trenutku u automatskom načinu rada robot izgubi „SafetyOk“ signal, smatra se da je došlo do pogreške na robotu te se rad cijelog sustava potpuno zaustavlja.



Slika 55. Logika za provjeru stanja Fanuc robota

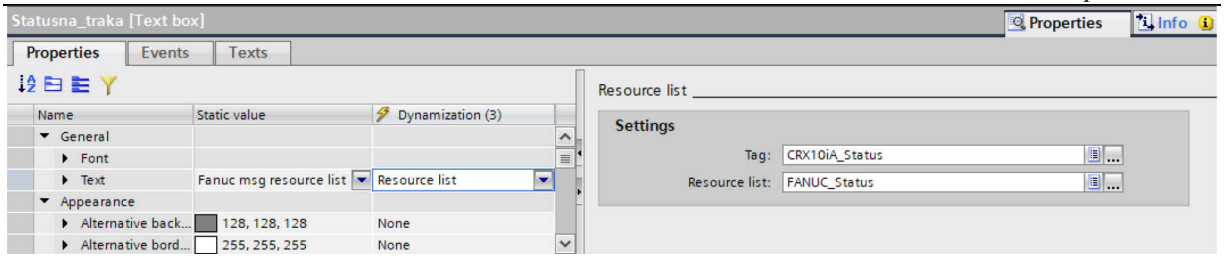
Za pokretanje, pauziranje i zaustavljanje robota, funkcijski blok uzima u obzir stanje sigurnosti sustava kao i njegov način rada te zavisno o njima upravlja s mogućnošću pokretanja robota u automatskom načinu rada [Slika 56].



Slika 56. Logika za upravljanje Fanuc robotom u automatskom načinu rada sustava

3.3.1.4. Programiranje na strani HMI-a

Za definiranje robotskog zaslona, korišten je prethodno definirani standardizirani zaslon robota [Slika 36] na kojeg su dodani već definirani *faceplate*-ovi. Jedino tipkalo koje nije *faceplate* je tipkalo za slanje robota u početni položaj, koje otvara standardizirani skočni prozor upozorenja [Slika 38]. WinCC Unified V16 ima pogrešku koja onemogućuje pravilno stvaranje *faceplate*-a za otvaranje skočnog prozora, stoga se u tom slučaju koristilo standardno tipkalo. Osim tipkala, potrebno je definirati i statusnu traku te ju povezati sa statusnim *integerom* kojim upravlja funkcijski blok robota na PLC-u [Slika 57].



Slika 57. Definiranje statusne trake za Fanuc CRX-10iA robota

Nakon potpunog definiranja zaslona robota dobije se sljedeći izgled [Slika 58].

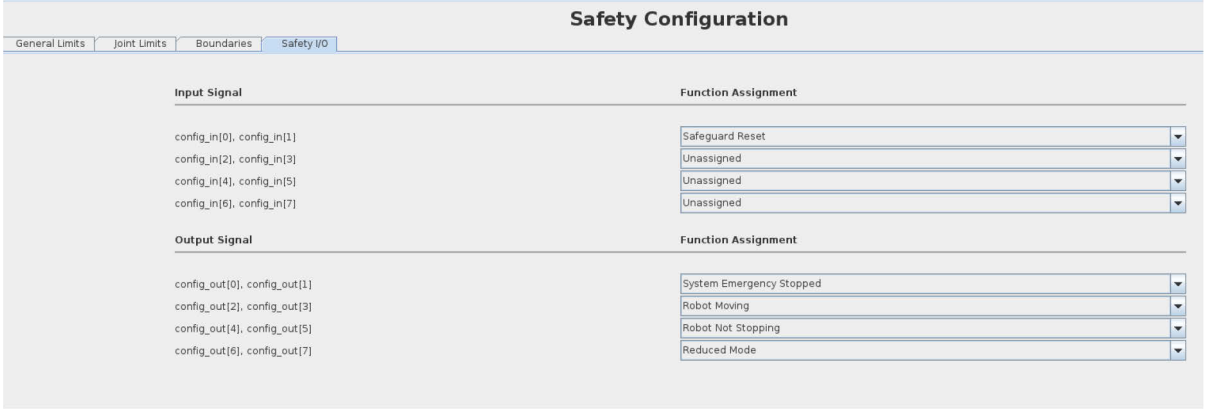
Slika 58. Potpuno definirani zaslon Fanuc CRX-10iA robota

3.3.2. UR kolaborativni robot

Kao što je bio slučaj kod Fanuc robota, UR roboti imaju međusobno identičan način rada te ih nije potrebno zasebno opisivati. Iz tog razloga, u nastavku će se obrađivati samo programska podrška za UR3 robota te će biti naglašeno ukoliko postoji razlika vezana za UR5 robot. Kod UR robota također nije bilo potrebe za korištenjem dodatnih programskih alata, već su sve postavke na robotu namještene pomoću robotskog *teach pendant*-a.

3.3.2.1. Programiranje na strani robota

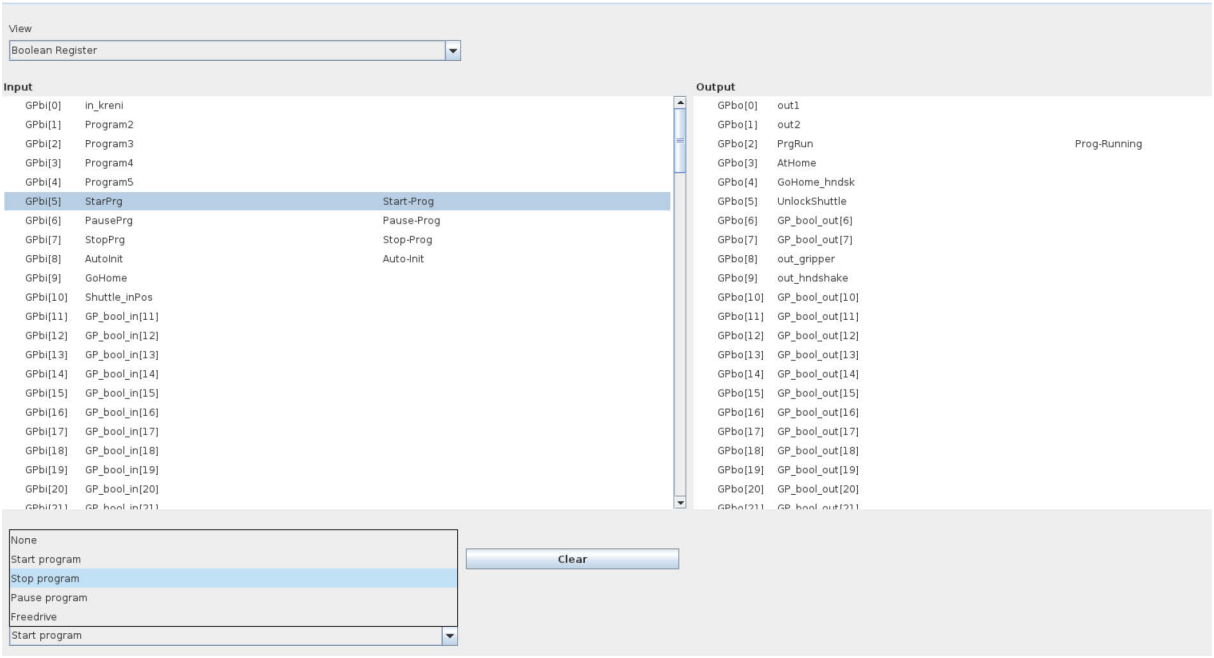
Prije programiranja, potrebno je konfigurirati komunikaciju s PLC-om, što se obavlja na „Installation“ zaslonu *teach pendant*-a. Pomoću njega moguće je pristupiti „Safety Configuration“ zaslonu [Slika 59] gdje je potrebno namjestiti sistemske ulazne i izlazne signale nužne za kontrolu robota pomoću PLC-a. Za pristup „Safety Configuration“ zaslonu potrebna je lozinka, koja je po standardu „ur“.



Slika 59. Konfiguracija sistemskih ulaznih i izlaznih signala na UR robotima

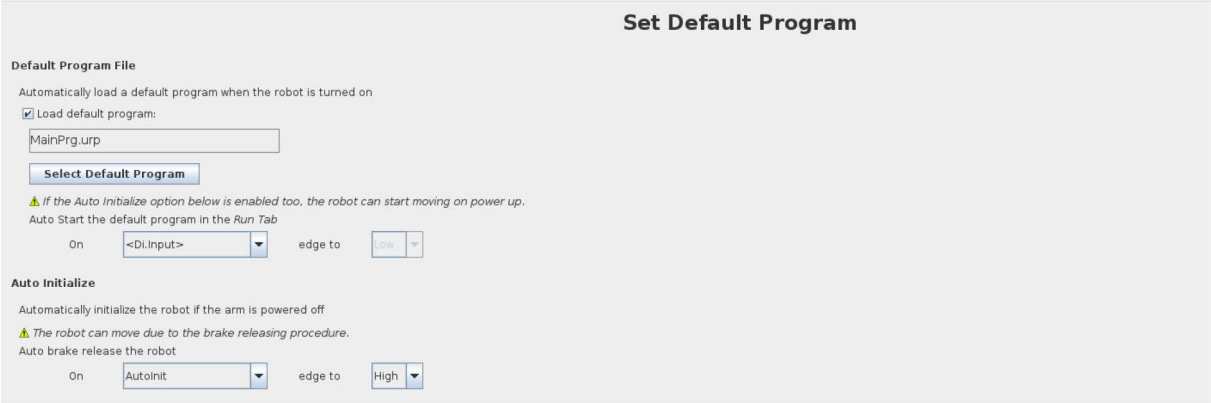
Na zaslonu „I/O Setup“ moguće je određenim varijablama pridodati akciju koja će se izvršiti nakon što se promijeni vrijednost varijable u logičku jedinicu [Slika 60]. Zato je potrebno određenim varijablama pridodati akcije kojima će PLC upravljati glavnim programom robota. Akcije koje se mogu pridodati varijablama su:

- Start program
- Stop program
- Pause program
- Freedrive



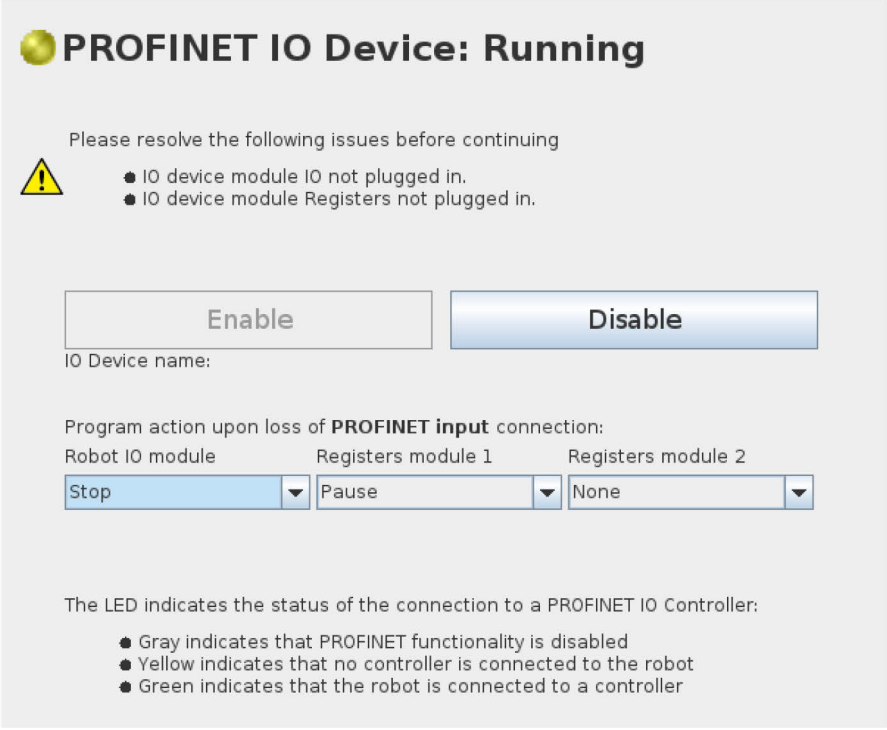
Slika 60. Konfiguracija varijabli za daljinsko upravljanje UR robota

Na zaslonu „Default Program“ potrebno je odabrati glavni program koji će se izvršavati pomoću prethodno definiranih varijabli [Slika 61]. Također je u ovom zaslonu moguće definirati ulazani signal „AutoInit“ koji će vršiti inicijalizaciju robota.



Slika 61. Odabir glavnog programa i ulaznog signala za inicijalizaciju UR robota

Na zaslonu „PROFINET“ potrebno je konfigurirati PROFINET vezu za komunikaciju s PLC-om. Ukoliko je komunikacija s PLC-om postignuta, na PROFINET zaslonu bi pod „PROFINET IO Device“ trebao pisati status „Connected“.



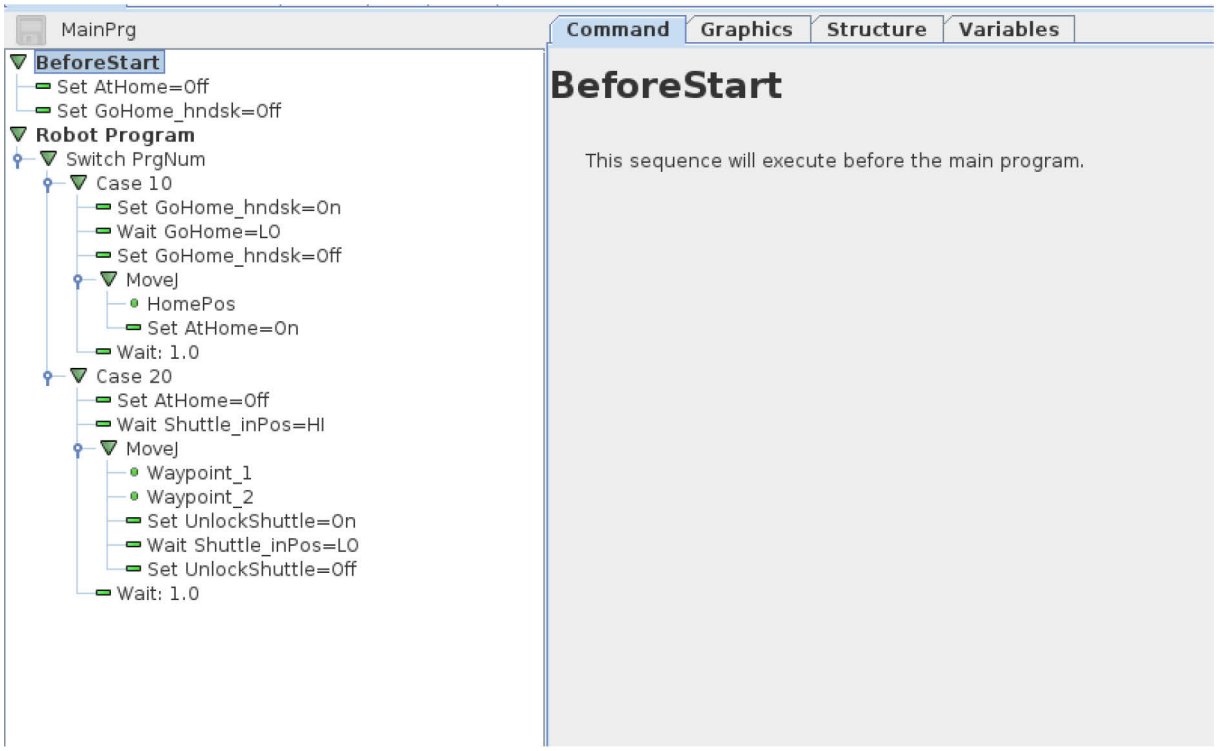
Slika 62. Konfiguracija PROFINET-a na UR robotu

UR5 robot ima dodatnu opciju na „Safety“ zaslonu naziva „Safe Home“. Njome je moguće definirati početni položaj robota u koji će robot otići svaki put kada se pokrene njegova inicijalizacija. UR3 nema tu opciju već je odlazak u početni položaj riješen pomoću glavnog programa. Glavni program, naziva „MainPrg“ [Slika 63], zapravo je *switch – case* program

gdje robot očekuje od PLC-a *integer* koji poziva izvršavanje određenog programskog slučaja. Tako se unutar programa nalaze dva definirana slučaja:

- Slučaj 10 – *homing* program
- Slučaj 20 – program s proizvodnim procesom robota

Prije izvršavanja *switch – case* programa, moguće je definirati sekvencu inicijalizacije „BeforeStart“. Unutar nje definirano je resetiranje svih signala koji se kasnije koriste u programu.

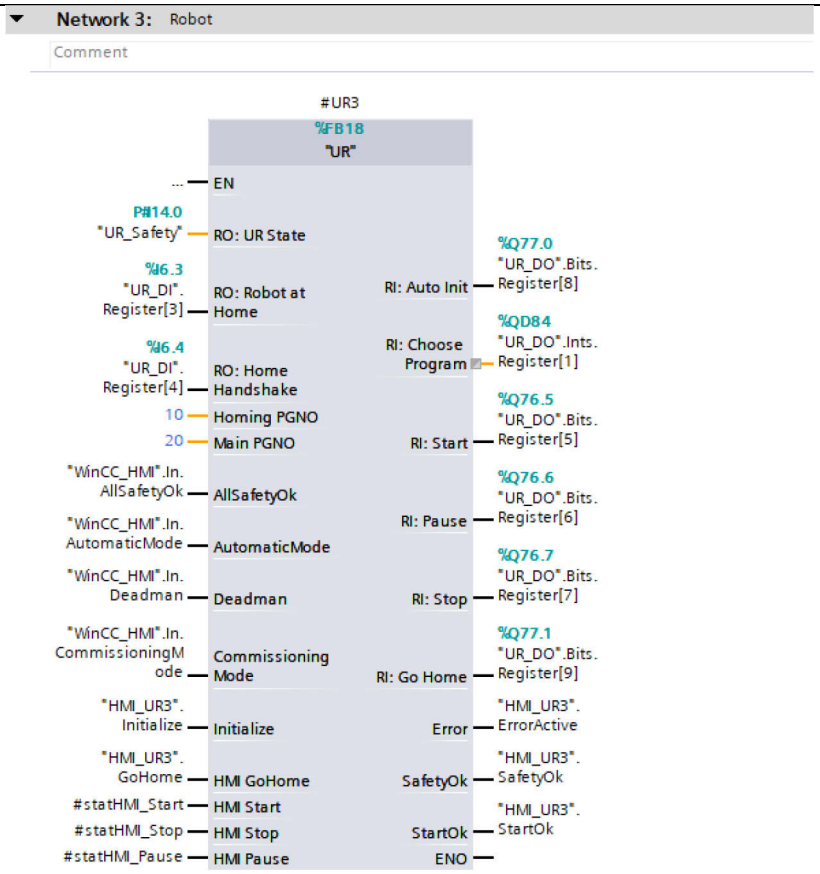


Slika 63. Definirani glavni program UR robota

Definiranjem glavnog programa završava se programiranje na strani robota. UR5 za razliku od UR3 robota očekuje još uključivanje „Remote“ načina rada za kontrolu robota pomoću PLC-a.

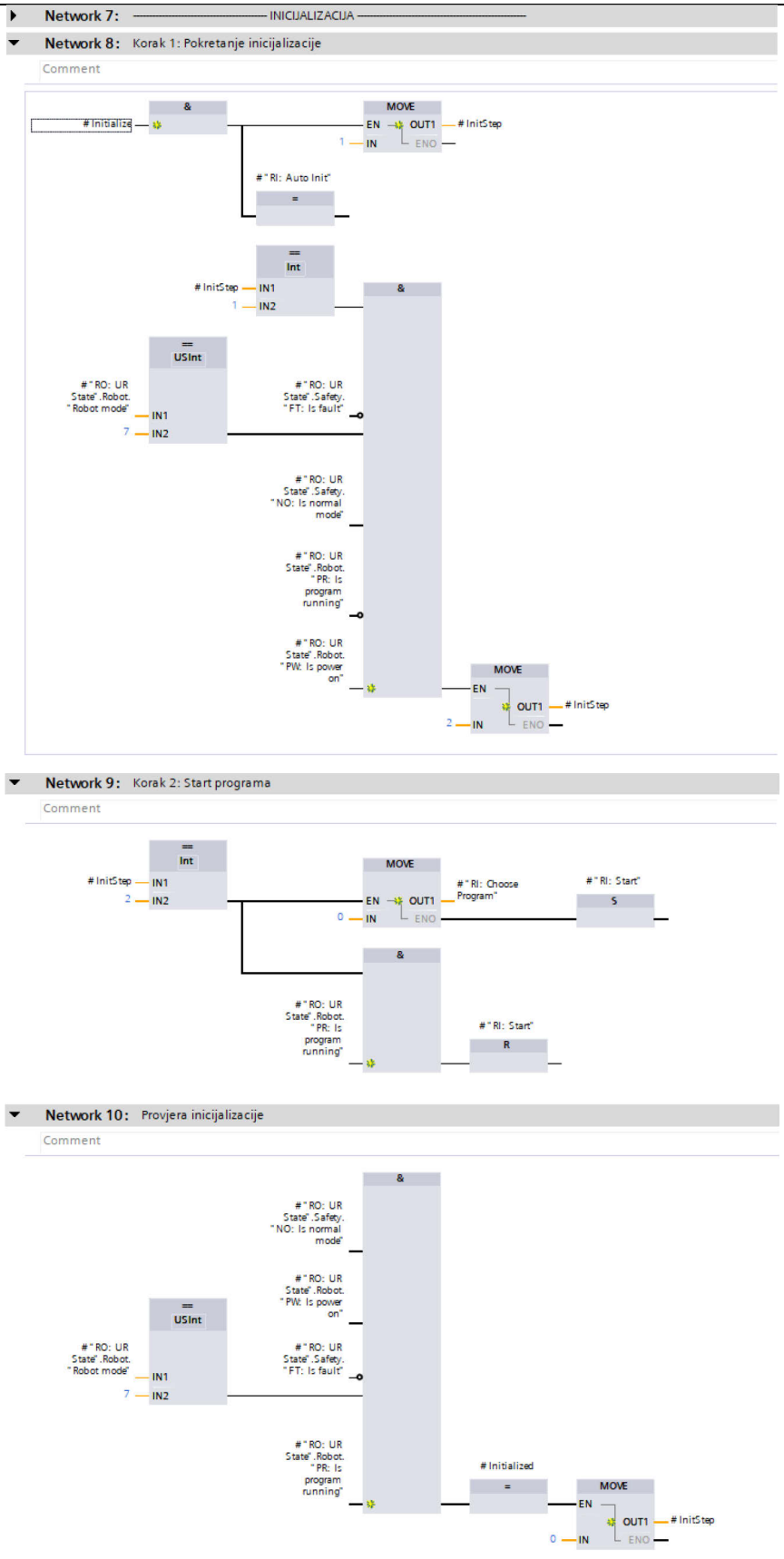
3.3.2.2. Programiranje na strani PLC-a

Na strani PLC-a ponovno je definiran standardizirani funkcijski blok [Slika 64]. Na ulazu od funkcijskog bloka također je moguće definirati drugačije vrijednosti programskih slučajeva, ukoliko dođe do izmjena u glavnom programu robota.



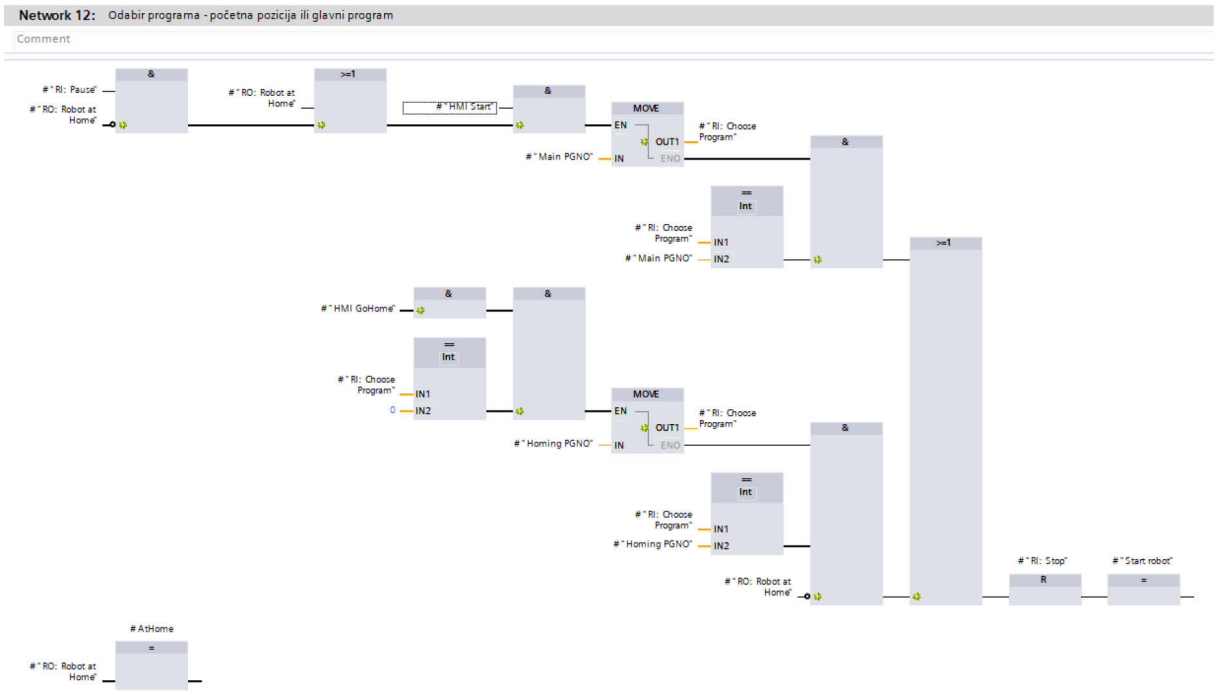
Slika 64. Funkcijski blok UR robota

Organizacija logike unutar bloka identična je već opisanoj organizaciji funkcijskog bloka Fanuc robota. Glavna razlika je u načinu inicijalizacije robota, s obzirom da se kod UR robota inicijalizacijom želi pokrenuti i glavni program u svrhu čega je isprogramirana kratka sekvenca za izvršavanje inicijalizacije [Slika 65]. Inicijalizacija se sastoji od dva koraka gdje se u prvom robotu šalje „AutoInit“ signal nakon čega se čeka paljenje motora i otpuštanje kočnica na robotu. Zatim se prelazi u drugi korak koji pokreće glavni program te robot očekuje broj programskog slučaja koji se želi izvršiti. Nakon inicijalizacije izvršava se provjera uspješne inicijalizacije koja resetira sekvencu inicijalizacije u početni korak.



Slika 65. Sekvenca za inicijalizaciju UR robota

Nakon inicijalizacije potrebno je odabrati željeni programski slučaj[Slika 66]. Uvijek je nakon inicijalizacije potrebno prvo izvršiti *homing* program. Potrebno je naglasiti da UR5 robot ima integriranu funkciju vraćanja u početni položaj prilikom svoje inicijalizacije. Pošto je nemoguće definirati putanju kojom će se robot vratiti u početni položaj, treba imati na umu opasnost od kolizije.



Slika 66. Odabir programskog slučaja na UR robotu

Postupak pokretanja, pauziranja ili zaustavljanja robota identičan je postupku opisanog na Fanuc robotima.

3.3.2.3. Programiranje na strani HMI-a

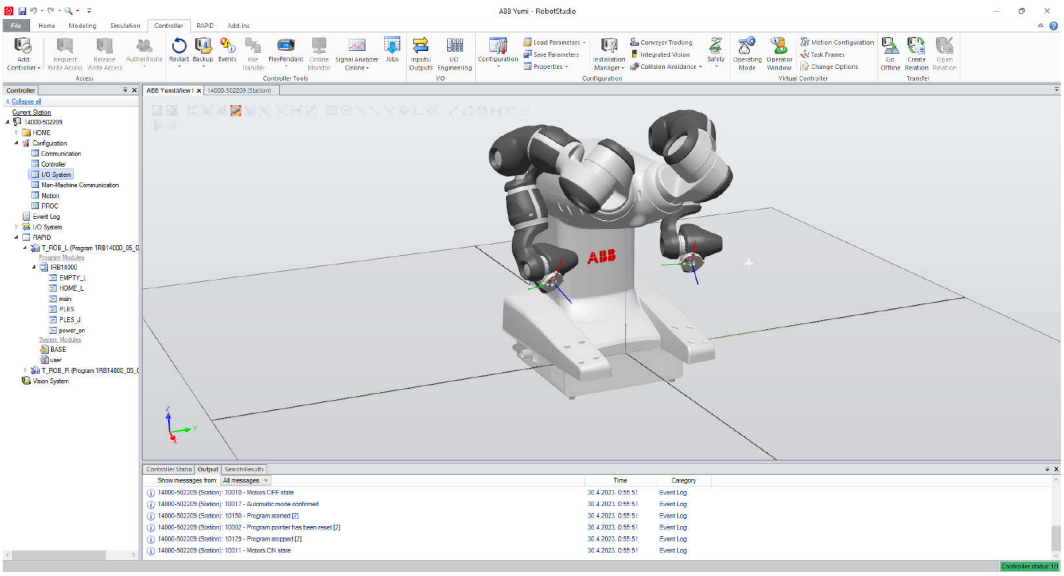
Kao i kod Fanuc robota, potrebno je standardiziranom zaslonu za robote dodati željene *faceplate*-ove, definirati njihove ulazne signale te definirati skočni zaslon za upozorenje robota pri kretanju u početni položaj. Nakon potpunog definiranja, robotski zaslon poprima oblik prikazan na [Slika 67].

Slika 67. Potpuno definirani zaslon UR3 robota

3.3.3. ABB kolaborativni robot

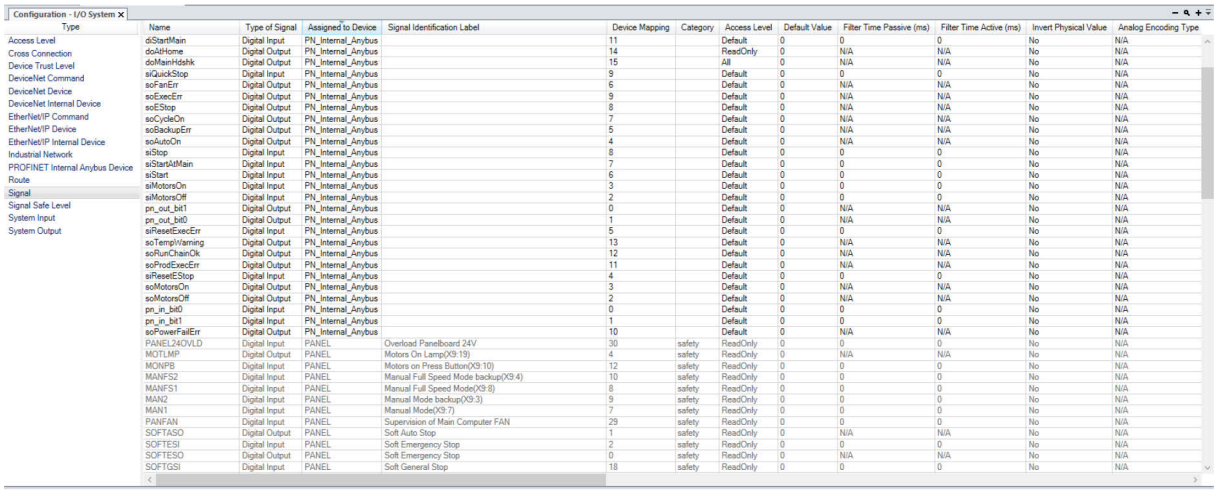
3.3.3.1. Korištenje RobotStudio programskog alata

Za programiranje i konfiguriranje ABB YuMi robota korišten je ABB-ov službeni programski alat naziva RobotStudio [Slika 68]. Alat omogućuje offline, ali i online programiranje robota, kao i direktno upravljanje robota pomoću računala koristeći „FlexPendant Viewer“ alat ukoliko je računalo direktno spojeno mrežnim kablom na servisni port robota. Alat je vrlo intuitivan za koristiti, a pri izradi ovog rada prvenstveno se koristio za konfiguraciju ulaznih i izlaznih signala na robotu.



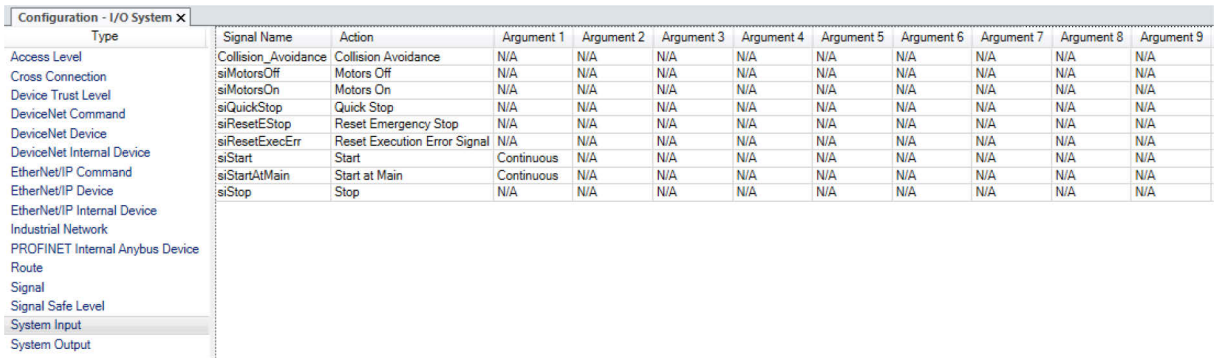
Slika 68. Sučelje ABB-ovog programskog alata RobotStudio

Za konfiguraciju signala, potrebno je pristupiti „I/O System“ zaslonu u RobotStudio-u. Odabirom kartice „Signal“ otvara se prikaz svih definiranih signala te njihovih informacija, poput uređaja kojem su dodijeljeni te lokacije signala. Ovdje je bilo potrebno definirati niz sistemskih ulaznih i izlaznih signala te ih mapirati na željenu lokaciju na PROFINET uređaju „PN_Internal_Anybus“ [Slika 69]. Mapiranje signala mora odgovarati konfiguraciji signala na strani PLC-a.



Slika 69. Definiranje ulaznih i izlaznih signala za ABB YuMi robota

Nakon stvaranja signala, svakom je sistemskom signalu potrebno pridijeliti sistemsku akciju koje će se izvršavati kada se promijeni vrijednost signala. Definirane su akcije iskorištene za daljinsko upravljanje robotom pomoću PLC-a. Ovo je jedina izmjena u konfiguraciji robota koja je potrebna prije izrade programske podrške na strani robota.

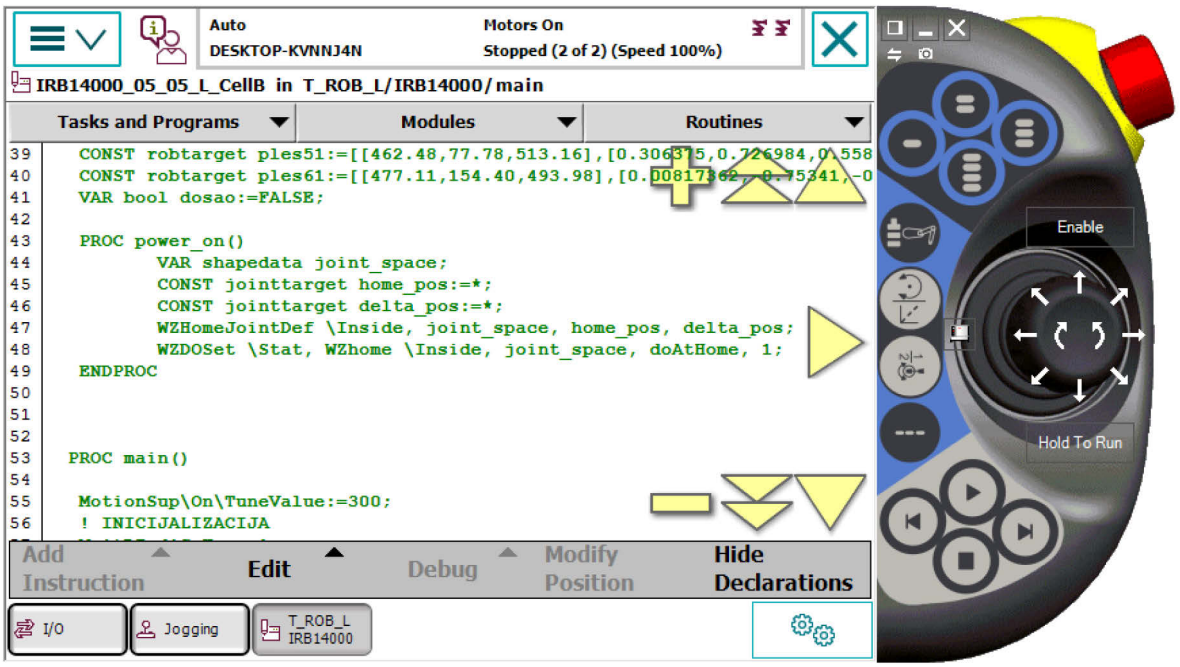


Slika 70. Pridjeljivanje akcija sistemskim signalima ABB YuMi robota

3.3.3.2. Programiranje na strani robota

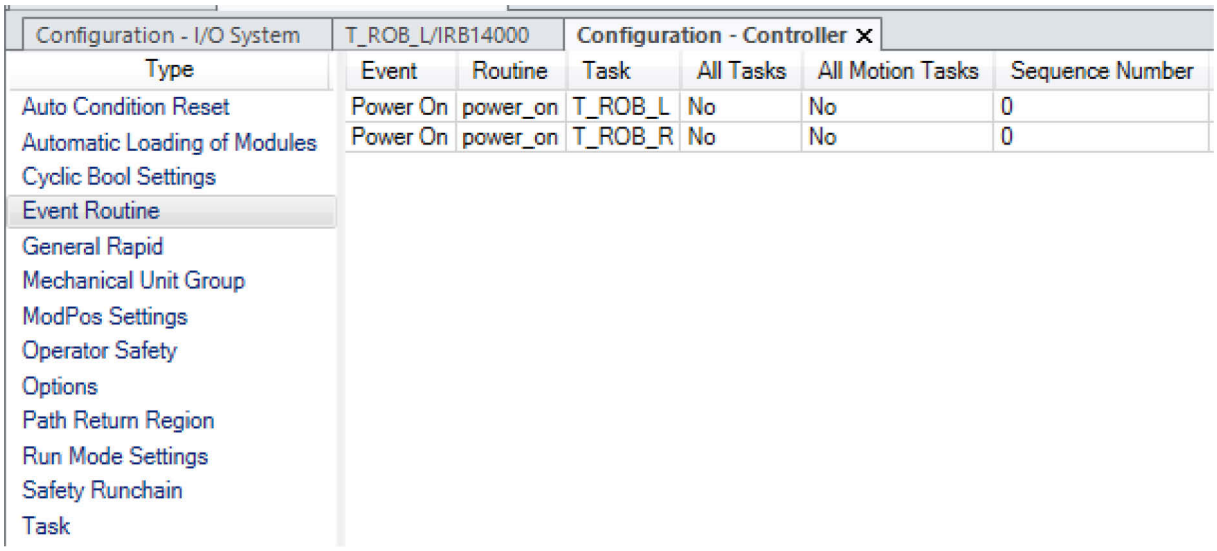
ABB robotski program sastoji se od po jednog glavnog programa po robotskoj ruci, koji sadrži rutine koje se mogu, ali i ne moraju pozivati u glavnoj petlji. Prije glavne petlje potrebno je definirati rutinu „power_on“ koja će se izvršiti samo prilikom paljenja robota. Unutar rutine

nalazi se programski kod koji definira početni položaj robota s dozvoljenim odstupanjem te signal koji će se paliti ukoliko se robot nađe u početnom položaju[Slika 71].



Slika 71. Definiranje početnog položaja ABB YuMi robota

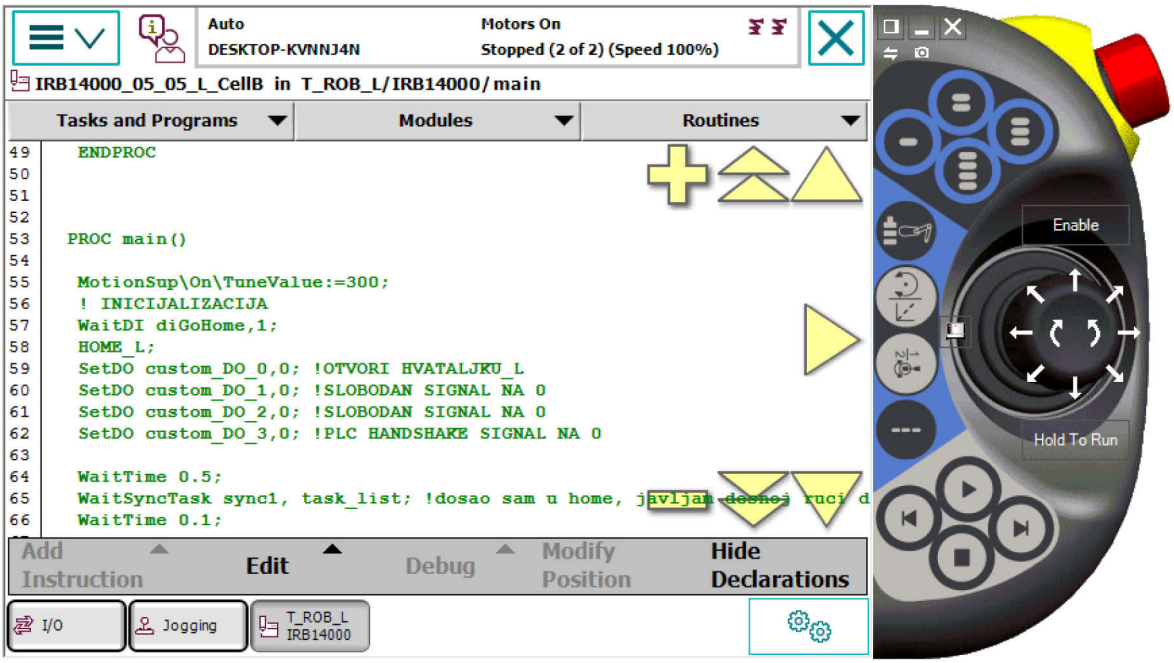
Potrebno je naglasiti da ABB YuMi ima dvije robotske ruke te isto tako i dva glavna programa, stoga je identičnu rutinu također potrebno definirati u glavnom programu druge ruke ABB YuMi robota. Navedenim „power_on“ rutinama je pomoću RobotStudio-a potrebno i pridijeliti „Power On“ događaj izvršavanja [Slika 72].



Slika 72. Pridjeljivanje „Power On“ eventa „power_on“ rutinama u RobotStudio-u

Na početku glavne petlje robot čeka signal za odlazak u početni položaj koji poziva *homing* rutinu. Time se osigurava da robot uvijek prvo zauzme početni položaj prije nego što prijeđe

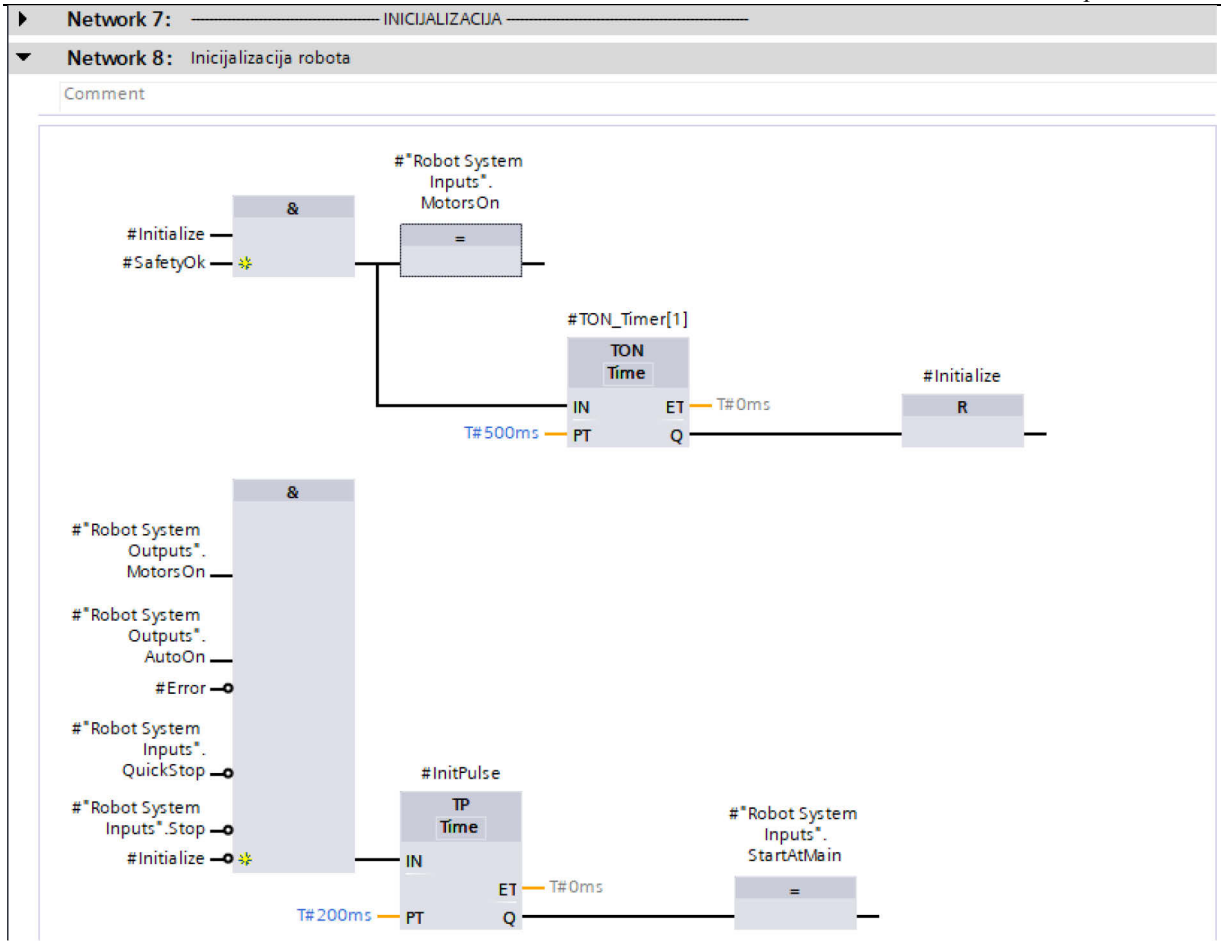
na izvršavanje proizvodnog procesa [Slika 73]. Ovime se završava nadogradnja robotskog programa. Zadnja postavka koju je potrebno promijeniti je postavljanje robota iz ručnog u automatski način rada.



Slika 73. Pozivanje *homing* rutine na početku glavne petlje ABB YuMi robota

3.3.3.3. Programiranje na strani PLC-a

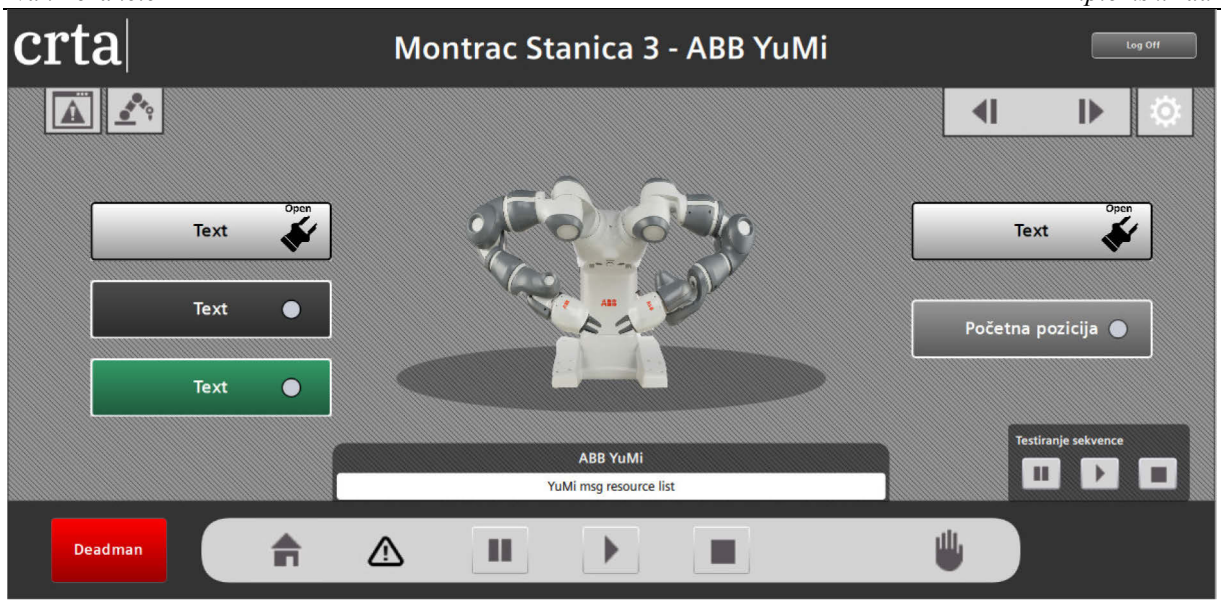
Organizacija logike unutar standardiziranog ABB funkcijskog bloka identična je organizaciji prethodno opisanih funkcijskih blokova preostalih robota. Potrebno je napomenuti da ABB roboti nemaju mogućnost resetiranja svih pogreški pomoću PLC-a te je u nekim slučajevima ipak nužno koristiti ABB-ov „FlexPendant“ za resetiranje. Postupak inicijalizacije robota [Slika 74] pokreće se pritiskom tipkala za inicijalizaciju na HMI-u kojim se pomoću funkcijskog bloka robota šalje signal za paljenje motora. Ukoliko su motori upaljeni te ukoliko je robot u automatskom načinu rada bez aktivnih grešaka, PLC daje impuls za paljenje glavnog robotskog programa, nakon kojeg robot ulazi u *homing* rutinu te čeka da se robot preko HMI-a pošalje u početni položaj. Postupak pokretanja, pauziranja ili zaustavljanja robota identičan je postupku opisanog na prethodnim robotima.



Slika 74. Logika za inicijalizaciju ABB robota

3.3.3.4. Programiranje na strani HMI-a

Postupak izrade zaslona robota identičan je kao i kod prethodnih robota. Jedina razlika je u tome što stanica s ABB YuMi robotom posjeduje dodatni cilindar koji služi za upresavanje, odnosno sklapanje dijelova proizvoda te je iz tog razloga dodan dodatni „ToggleButton“ *faceplate* kojim se može testirati rad preše u ručnom načinu rada sustava. Konačni zaslon ABB YuMi robota poprima izgled prikazan na [Slika 75].

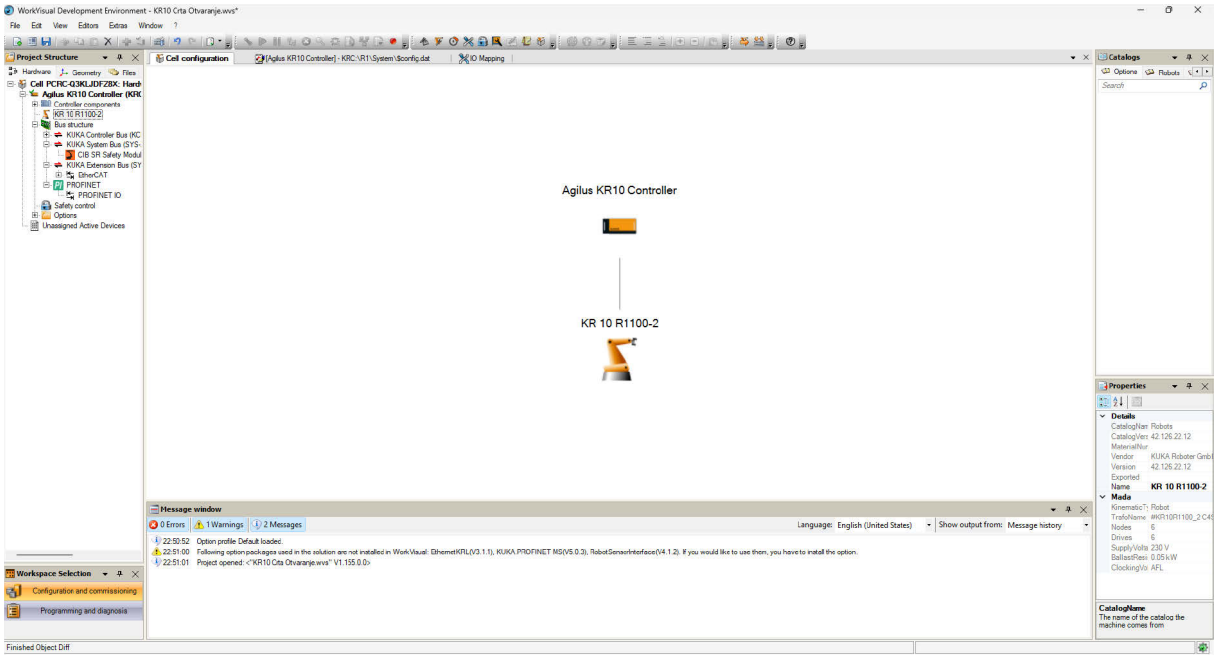


Slika 75. Potpuno definirani zaslon ABB YuMi robota

3.3.4. KUKA industrijski robot – Agilus KR10 R1100-2

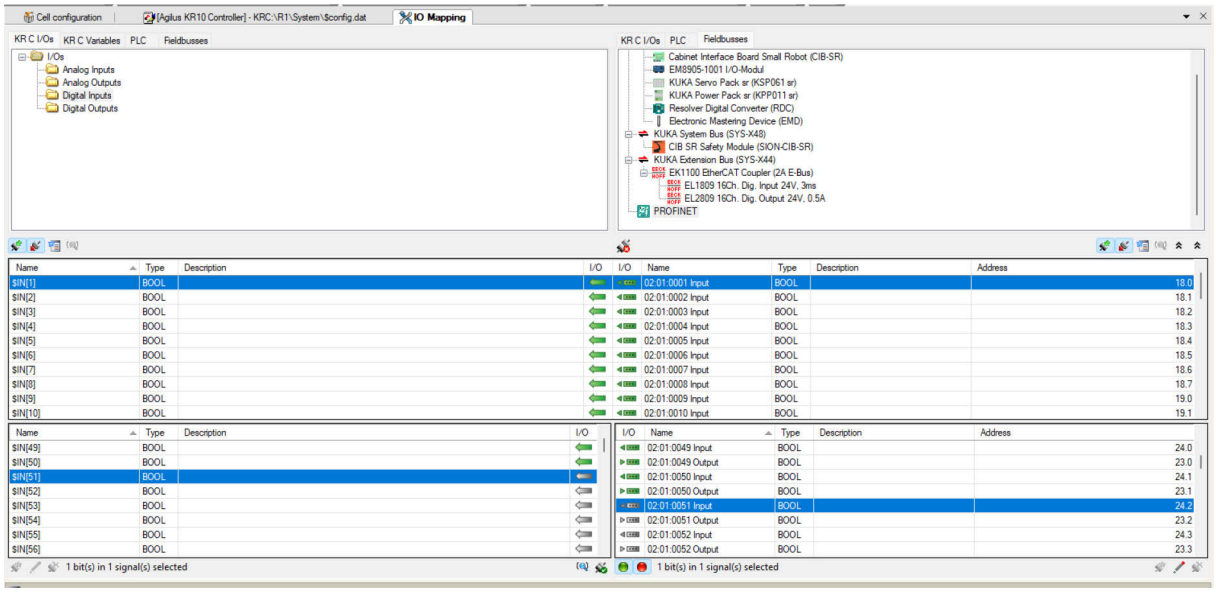
3.3.4.1. Korištenje WorkVisual programskog alata

Za programiranje i konfiguriranje KUKA Agilus robota korišten je programski alat tvrtke KUKA naziva WorkVisual [Slika 76].



Slika 76. Sučelje KUKA WorkVisual programskog alata

Prije programiranja, potrebno je mapirati digitalne ulazne i izlazne signale robota na PROFINET te time omogućiti komunikaciju s PLC-om. Mapiranje se izvodi na „I/O Mapping“ zaslonu gdje je signale od robota potrebno mapirati s PROFINET signalima. Mapirani signali su na zaslonu označeni zelenom, dok su nemapirani signali označeni sivom bojom [Slika 77].



Slika 77. Mapiranje ulaznih i izlaznih signala KUKA Agilus robota

KUKA roboti razlikuju tri vrste robotskih datoteka s različitim ekstenzijama:

1. .src ekstenzija – označuje datoteku programa kojeg robot izvršava kada se program pozove
2. .dat ekstenzija – označuje podatkovnu datoteku, često povezanu s datotekom programa
3. .sub ekstenzija – označuje program koji se izvršava kao pozadinski zadatak

Programi, bilo .src ili .sub ekstenzije, podijeljeni su u segmente koji se još nazivaju *folds*. Programi se izvršavaju sekvencijalno, a *folds* segmenti služe za pojednostavljeno upravljanje istima korištenjem „SmartPAD“ *teach pendant*-a robota. KUKA roboti sadrže predefinirane datoteke na kojima je često potrebno napraviti tek nekoliko izmjena za daljinsko upravljanje robotom PLC-om. Prve takve izmjene napravljene su na sistemskoj podatkovnoj datoteki „$config.dat“ gdje su korištenim signalima pridružena imena kako prilikom programiranja operater ne mora znati točnu lokaciju korištenih signala već samo njihov naziv. Definirani su signali za upravljanje odabranim programom robota u automatskom načinu rada te izlazni signali koji upućuju na trenutno stanje robotskog programa [Slika 78].

[Agilus KR10 Controller] - KRC:\R1\System\$config.dat | [Agilus KR10 Controller] - KRC:\R1\System\sps.sub

DEF_OV_PRO(INT)

```
805   CONV_PART_NBR[9]=0
806   CONV_PART_NBR[10]=0
807   CONV_PART_NBR[11]=0
808   CONV_PART_NBR[12]=0
809   CONV_PART_NBR[13]=0
810   CONV_PART_NBR[14]=0
811   CONV_PART_NBR[15]=0
812   CONV_PART_NBR[16]=0
813
814   ENUM CONV_MODE_TYPE Absolute,Relative
815   DECL CONV_MODE_TYPE ActualConveyorMode=#Relative
816 ;ENDFOLD Conveyor
817 ;FOLD USER GLOBALS
818 ;*****************************************
819 ;Digital outputs
820 SIGNAL progStarted $OUT[10]
821 SIGNAL progPaused $OUT[11]
822 SIGNAL progEnd $OUT[12]
823
824 ;Digital inputs
825 SIGNAL startProg $IN[10]
826 SIGNAL pauseProg $IN[11]
827 SIGNAL cancelProg $IN[12]
828 SIGNAL resumeProg $IN[13]
829 ;Make your modifications -ONLY- here
830 ;*****************************************
831 ;==================================
```

Slika 78. Definiranje signala za upravljanje programom KUKA robota u automatskom načinu rada

Sljedeće izmjene napravljene su unutar „cell.src“ programa kojeg je nužno koristiti za udaljeno upravljanje KUKA robotom. Navedeni program služi kao glavni program koji se pokreće prilikom inicijalizacije robota s HMI-a te radi na principu *switch – case* logike. Unutar programa definirana su dva programska slučaja, *homing* program te program proizvodnog procesa [Slika 79]. Za pauziranje programa preko HMI-a, potrebno je definirati „pause“ program koji će se pozivati kao prekidni (eng. *interrupt*) program unutar „cell“ programa. Program za pauziranje koristi „BRAKE“ funkciju koja prekida izvođenje programa sve dok je signal za pauziranje aktivan. Čim je signal uklonjen, program nastavlja s radom [Slika 80].

CELL ( )

```
;EXT EXAMPLE3 ( )

;FOLD INIT
DECL CHAR DMY[3]
DMY[]="---"
;ENDFOLD (INIT)
;FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )

GLOBAL INTERRUPT DECL 11 WHEN pauseProg==TRUE DO pause()
INTERRUPT ON 11
;ENDFOLD (BASISTECH INI)
;FOLD AUTOEXT INI
P00 (#INIT_EXT,#PGNO_GET,DMY[],0 ) ; Initialize extern mode
;ENDFOLD (AUTOEXT INI)
LOOP
  P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
  SWITCH  PGNO ; Select with Programnumber

  CASE 2
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    ;EXAMPLE2 ( ) ; Call User-Program

  CASE 10
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    homing() ; Call User-Program

  CASE 20
    P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
    crta_otvaranje(); Call User-Program

  DEFAULT
  ENDSWITCH
ENDLOOP
END
```

Slika 79. Definiranje *switch – case* logike unutar „cell“ programa KUKA robota

pause ( )

```
&ACCESS RVP1
&REL 3
&PARAM EDITMASK = *
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM DISKPATH = KRC:\R1\Program
DEF pause( )
BRAKE F

progPaused=TRUE ;program paused output


WAIT FOR NOT pauseProg ;pause program input
progPaused=FALSE
END
```

Slika 80. Definiranje „pause“ programa KUKA robota

Kao proizvodni proces odabran je prethodno definirani program „crta_otvaranje.src“ unutar kojeg je jedina promjena bilo deklariranje prekidnog programa „cancelling“ koji prekida trenutni aktivni slučaj te vraća robota u „cell“ program, gdje robot čeka pozivanje novog programskog slučaja.

Kako su sve programske promjene napravljene koristeći WorkVisual alat, na strani robota potrebno je samo mapirati sistemske signale za daljinsko upravljanje robota PLC-om. Signali se mapiraju na zaslonu „Automatic External Configuration“ na *teach pendant*-u robota [Slika 81].

Item	Description
1	Number
2	Long text name of the input/output

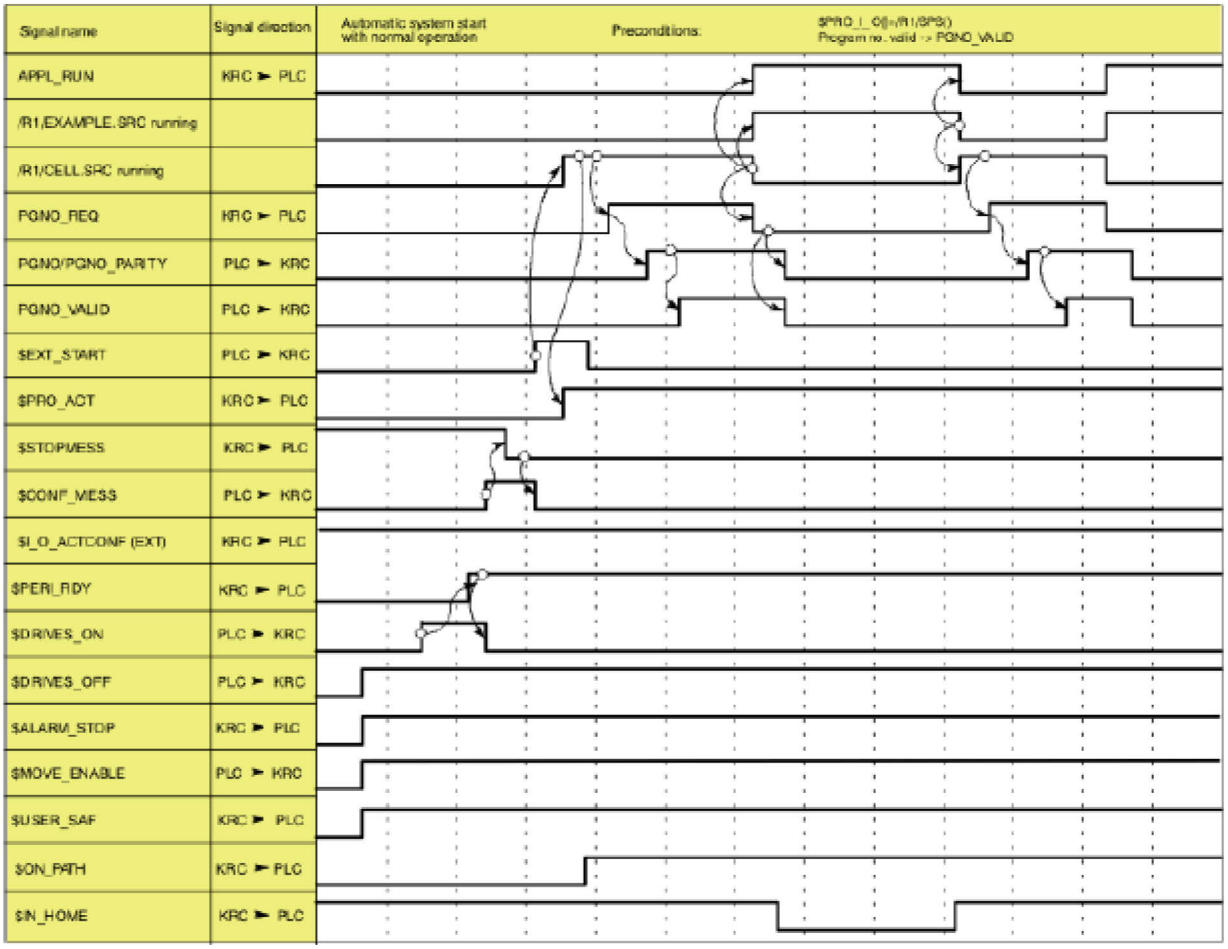
Item	Description
3	Type ■ Green: Input/output ■ Yellow: variable or system variable ($...)
4	Name of the signal or variable
5	Input/output number or channel number
6	The outputs are thematically assigned to tabs.

Slika 81. Mapiranje signala za daljinsko upravljanje KUKA robotom [6]

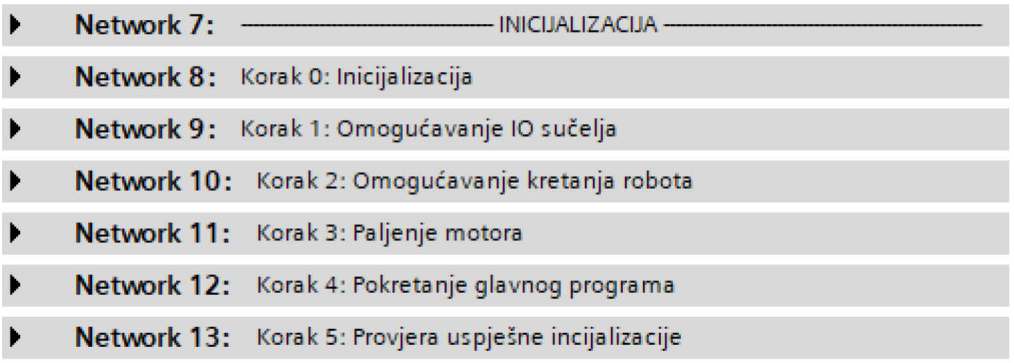
3.3.4.2. Programiranje na strani PLC-a

Na strani PLC-a napravljen je standardizirani funkcijski blok za KUKA robote prema uputama iz KUKA priručnika [6]. KUKA naime zahtjeva određen redoslijed paljenja signala za automatsko daljinsko upravljanje robotom [Slika 82], stoga je u tu svrhu napravljena sekvenca inicijalizacije koja se pridržava danih uputa [Slika 83]. Sekvenca započinje resetiranjem svih

sistemskih ulaza robota a završava provjerom uspješnog pokretanja „cell“ programa nakon čega robot čeka odabir programskog slučaja.

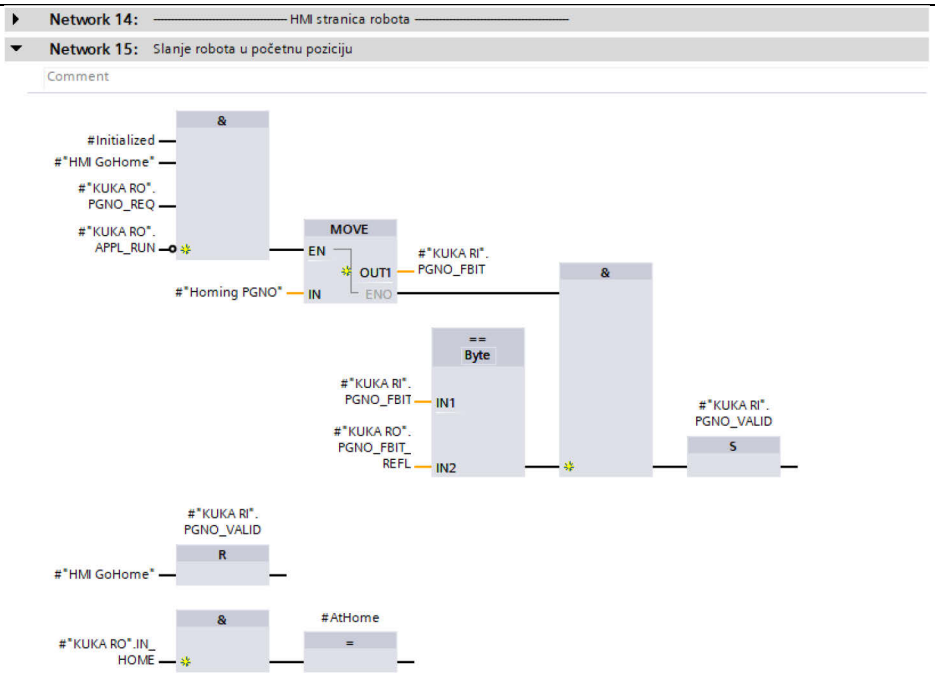


Slika 82. Dijagram signala za automatsko daljinsko upravljanje KUKA robotom [6]



Slika 83. Sekvenca za inicijalizaciju KUKA robota

Prilikom odabira programskog slučaja, robot vraća identičnu vrijednost slučaja nazad PLC-u te PLC mora obaviti usporedbu vrijednosti. Ukoliko su vrijednosti iste, došlo je do točnog odabira programa te PLC šalje robotu „PGNO_VALID“ signal [Slika 84].



Slika 84. Logika za odabir programskog slučaja vraćanja KUKA robota u početni položaj

Pokretanje robota u automatskom načinu rada postiže se opisanim postupkom slanja vrijednosti željenog programskog slučaja, dok se pauziranje i zaustavljanje programa postiže slanjem signala koji pokreću prekidne programe prethodno definirane na robotu.

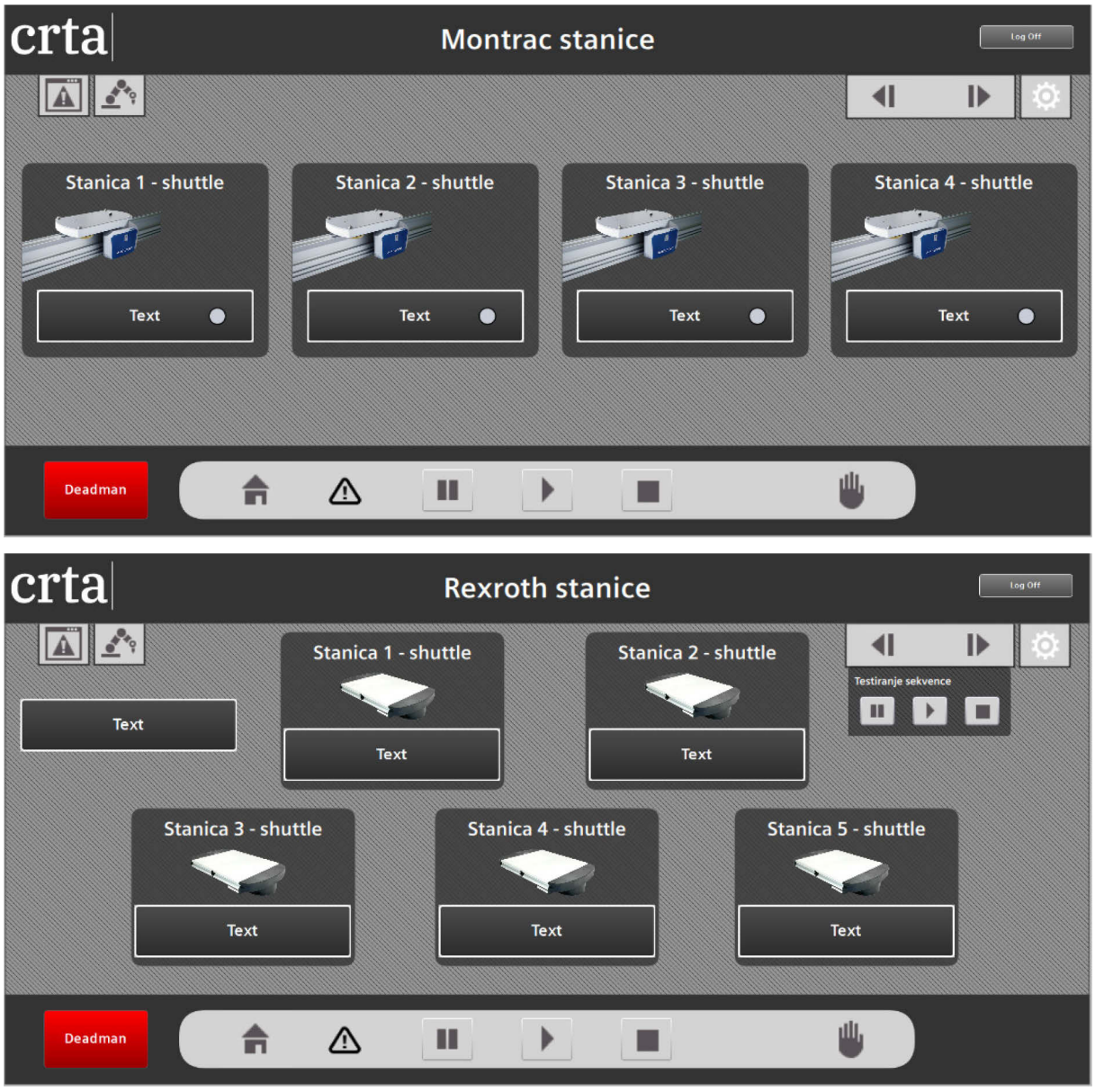
3.3.4.3. Programiranje na strani HMI-a

Na strani HMI-a, stvaranje zaslona provedeno je postupkom istovjetnim onima za preostale robote. Nakon dodavanja potrebnih *faceplate*-ova i tipkala dobije se sljedeći izgled zaslona za upravljanje KUKA Agilus robotom [Slika 85].

Slika 85. Potpuno definirani zaslon KUKA Agilus robota

4. TESTIRANJE I PUŠTANJE U POGON

Nakon što je napravljena potpuna konfiguracija sustava, potrebno je testirati spremnost sustava za automatski način rada te na kraju pustiti pogon u rad. Za testiranje svakog robota zasebno, korištena je prethodno opisana postavka za puštanje pogona u rad. Paljenjem navedene postavke sustav je i dalje u ručnom načinu rada, ali omogućava testiranje svakog robota zasebno ignorirajući sigurnost sustava u cijelosti, već gledajući samo sigurnost testirane stanice. Pošto su svi roboti konfigurirani tako da za pokretanje proizvodnog procesa čekaju signale s transportnog sustava, kreirana su dodatna dva HMI zaslona za upravljanje transportnim sustavima [Slika 86].

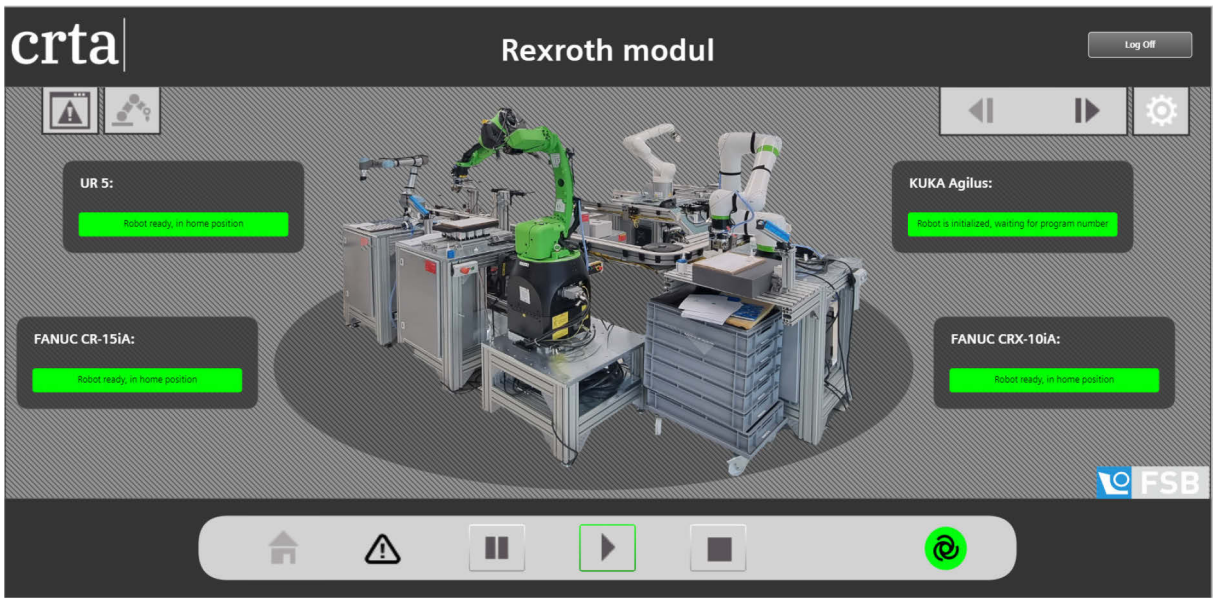


Slika 86. Definirani zaslona za upravljanje transportnim sustavima

Na taj način omogućeno je testiranje automatskog načina rada robota te usavršavanje funkcijskih blokova robota na strani PLC-a. Opcija „Način puštanja u rad“ pokazala se kao iznimno važna s obzirom da omogućuje znatnu fleksibilnost prilikom testiranja rada sustava. Tek kada je dokazana sigurnost svakog pojedinog robota, testirana je sigurnost sustava u automatskom načinu rada. Prije testiranja automatskog načina rada, potrebno je provjeriti jesu li zadovoljeni sljedeći uvjeti:

1. Oba PLC-a i njihovi moduli moraju biti uključeni
2. Svi roboti moraju biti uključeni
3. Na UR robotima mora biti odabran „MainPrg“ program
4. UR5 robot mora biti u „Remote“ načinu rada
5. Fanuc roboti moraju biti u „Auto“ načinu rada
6. Fanuc CRX-10iA mora imati potvrđen „Payload“
7. KUKA robot mora biti u „AUT EXT“ načinu rada te imati odabran „cell“ program
8. ABB YuMi robot mora biti u „Auto“ načinu rada

Kada su svi uvjeti zadovoljeni, potrebno je resetirati moguće postojeće pogreške u sustavu, inicijalizirati sve robote te sve robote poslati u početni položaj. Spremnost svih robota očituje se zelenom bojom statusne trake robota na početnim stranicama modula te zelenom bojom „Start“ tipkala [Slika 87]. Nakon što su svi roboti spremni, moguće je pokrenuti automatski način rada te pustiti pogon u rad.



Slika 87. Prikaz sustava spremnog za automatski način rada

5. MODULARNOST SUSTAVA

Kroz diplomski rad više puta je naglašena važnost standardizacije prilikom izrade TIA Portal projekta, a sada će biti objašnjeno i kako se standardizacija može iskoristiti za modularno unošenje promjena u sustavu. Industrijski se sustavi rijetko kad mijenjaju nakon puštanja u pogon, međutim ukoliko dođe do potrebe za istim standardizacija je čimbenik koji omogućava unos promjena bez potrebe za mijenjanjem osnovnih dijelova sustava. Kako svi roboti nakon standardizacije komuniciraju sa sustavom pomoću identičnih signala, moguće je u bilo kojem trenutku dodavati, uklanjati ili mijenjati robote po transportnim stanicama bez da se trajno naruši sigurnost rada sustava u cijelosti. Ukoliko bi željeli dodati novog robota u sustav na neku od slobodnih stanica, postupak izmjene sustava bi bio sljedeći:

1. Fizički postaviti robota na slobodnu stanicu te ga spojiti na podmrežu s ostatkom sustava
2. U TIA Portal projektu dodati standardizirani funkcijski blok odabranog robota te na njega spojiti robotske ulazne i izlazne signale, kao i signale za komunikaciju s korisničkim sučeljem
3. Dodati sigurnosne signale robota na „SigurnostSustava“ funkcijski blok
4. U HMI projektu stvoriti novi zaslon robota koristeći standardizirani zaslon
5. Spojiti statusnu i alarmnu listu robota s izlaznim varijablama funkcijskog bloka robota

Iz priloženog je vidljivo da proces izmjene postaje izuzetno jednostavan i brz što je direktan rezultat standardizacije sustava. U svrhu standardizacije napravljena je TIA Portal biblioteka podataka naziva „CRTA_Library“ koja u sebi sadrži sve funkcijske blokove, tipove podataka, *faceplate*-ove, HMI zaslone te HMI alarmne i tekst liste izrađene unutar ovog projekta. Ova biblioteka bi trebala poslužiti kao temelj CRTA standarda, kojeg će budući korisnici moći nadograđivati novim funkcijama te u kojem će unaprjeđivati već postojeće. Neki od prijedloga za nadogradnju sustava bili bi:

- Dodavanje nadzorne kamere koja bi omogućila emitiranje stanja sustava preko korisničkog sučelja
- Praćenje ciklusnog rada svake stanice te spremanje podataka o produktivnosti rada, kao i primjereni vizualizacijski prikaz
- Integracija značajki drugih web tehnologija za stvaranje bogatih vizualizacija poput animacija kretanja proizvoda unutar sustava

6. ZAKLJUČAK

Važan dio današnje proizvodnje je razvoj i unaprjeđenje automatizacije radi postizanja maksimalne iskoristivosti i fleksibilnosti industrijskih proizvodnih sustava, kao i podizanja sustava sigurnosti. Četvrtom industrijskom revolucijom došlo je do potrebe za razvojem proizvodnih sustava nove generacije s mogućnošću nadzora na daljinu pomoću vizualizacijski i funkcionalno naprednih korisničkih sučelja. Cilj ovog diplomskog rada bila je automatizacija i razvoj centralnog upravljanja industrijskim i kolaborativnim robotskim sustavom u Regionalnom centru izvrsnosti za robotske tehnologije (CRTA) koja vjerno predstavlja moderne industrijske proizvodne pogone. Tijekom rada postignuta je organizacija robotskog sustava u dva modula, od kojih je svaki organiziran oko zasebnog transportnog sustava. Svaki modul sadrži zaseban PLC kontroler zadužen za nadzor i kontrolu proizvodnog procesa cjelokupnog modula. Razvijeno je jedinstveno sučelje koje uklanja potrebu za tradicionalnim zasebnim sučeljima svakog modula, već omogućuje kontrolu i nadzor oba modula istovremeno, kao i svih uređaja unutar istih. Osim jedinstvenosti, sučelje sadrži prednosti i u modernoj i intuitivnoj vizualizaciji, jednostavnosti pristupa s različitih vrsta uređaja pomoću web preglednika, mogućnosti jednostavne nadogradnje i modularne izmjene. Modularnost sustava postignuta je standardizacijom osnovnih dijelova TIA Portal projekta koja je pretvorila trenutni sustav u temelj za buduće perspektive koji sustav mogu podići na još veću razinu.

LITERATURA

[1] »Siemens Global«. Dostupno: https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html, zadnji pristup: 23.04.2023.

[2] »EID Robotics«. Dostupno: https://eidrobotics.com/building-an-automated-assembly-line-how-to-minimize-costs-and-ramp-up-time/, zadnji pristup: 23.04.2023.

[3] »ABB«. Dostupno: https://new.abb.com/products/robotics/robots/collaborative-robots/yumi/irb-14000-yumi, zadnji pristup: 24.04.2023.

[4] »Wired Workers«. Dostupno: https://www.wiredworkers.io/cobot/brands/universal-robots/ur3/, zadnji pristup: 24.04.2023.

[5] »Robots Done Right«. Dostupno: https://robotsdoneright.com/Articles/comparing-fanuc-cr-and-crx-collaborative-robots.html, zadnji pristup: 25.04.2023.

[6] »KUKA System Software 8.2 Operating and Programming Instuctions for System Integrators« KUKA Roboter GmbH, Augsburg, Njemačka, 20.07.2012.

PRILOZI

I. ERROR_TO_REG.kl program za Fanuc robote

II. Sigurnosna kopija svih robota i TIA Portal projekta - https://github.com/IvanLozancic/DiplomskiRad/blob/main/backups.zip

III. CRTA biblioteka za TIA Portal projekte stvorena u svrhu standardizacije - https://github.com/IvanLozancic/DiplomskiRad/blob/main/CRTA_Library.zip

ERROR_TO_REG.kl program:

```
PROGRAM error_to_reg
%NOLOCKGROUP
%NOPAUSE= COMMAND + TPENABLE + ERROR
%NOBUSYLAMP
%NOABORT = ERROR + COMMAND
%COMMENT= 'ERROR SEND'



VAR
par_no,par_typ,par_int,int_reg:INTEGER
par_real:REAL
par_str:STRING[10]



seq_num,error_code,cause_code,time_int,severity,STATUS :INTEGER
error_string,cause_string,prog_nam:STRING[40]
prog_stop:BOOLEAN



BEGIN
par_no=1
GET_TPE_PRM(par_no,par_typ,par_int,par_real,par_str,STATUS)
IF (par_typ=1) AND (STATUS=0)THEN
int_reg=par_int
ENDIF



ERR_DATA(MAXINT, error_code, error_string, cause_code, cause_string,
time_int,severity, prog_nam)
SET_INT_REG( int_reg,error_code DIV 1000,STATUS)
SET_INT_REG( int_reg+1,error_code MOD 1000,STATUS)




END error_to_reg
```