

# Projektiranje modularnog proizvodnog sustava

---

Vico, Anđelko

Undergraduate thesis / Završni rad

2019

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:700216>

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

Download date / Datum preuzimanja: **2025-03-13**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Andelko Vico**

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentori:

Prof. dr. sc. Joško Petrić

Student:

Anđelko Vico

Zagreb, 2019.

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

Zahvaljujem se svom mentoru prof. dr. sc. Jošku Petriću za vodstvo i savjetovanje, zaposlenicima tvrtke Festo, Jasminki Poje i Antoniu Radiću za svu pruženu pomoć, te profesorima Tehničke škole Ruđera Boškovića, Zoranu Kauzlariću, Goranu Nuskrenu, Ervinu Palanoviću, Nevenu Malešu na ukazanom povjerenju i pruženoj pomoći tijekom izrade završnog rada.

Andelko Vico



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



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

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

## ZAVRŠNI ZADATAK

Student: **Anđelko Vico** Mat. br.: 0035201621

Naslov rada na hrvatskom jeziku: **Projektiranje modularnog proizvodnog sustava**

Naslov rada na engleskom jeziku: **Modular production system design**

Opis zadatka:

Modularni proizvodni sustav (MPS) firme FESTO sastoji se od četiri radne stanice te robota Mitsubishi Movemaster RV-M1. Svaka stanica koristi svoj programabilni logički kontroler (PLC) za upravljanje te sigurnosne sklopke, releje i napajanje. Svaka radna stanica je neovisna, ali mogu raditi i povezano tako da prethodna daje naredbu sljedećoj za daljnji rad. Radne stanice imaju zadatak pripremiti obradak, izmjeriti ga i sortirati, strojno ga obraditi te složiti. Robot prenosi obradak s jedne radne stanice na drugu.

U zadatku je potrebno projektirati i sastaviti MPS tako da može obavljati zadani radni zadatak, povezati elektroničke i pneumatske komponente te programirati upravljanje radnim stanicama i robotom. Po mogućnosti potrebno je napraviti i vizualizaciju modularnog proizvodnog sustava koristeći prikladno programsko okruženje.


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


Zadatak zadan:  
29. studenog 2018.

Rok predaje rada:  
1. rok: 22. veljače 2019.  
2. rok (izvanredni): 28. lipnja 2019.  
3. rok: 20. rujna 2019.

Predviđeni datumi obrane:  
1. rok: 25.2. - 1.3. 2019.  
2. rok (izvanredni): 2.7. 2019.  
3. rok: 23.9. - 27.9. 2019.

Zadatak zadao:

  
Prof. dr. sc. Joško Petrić

Predsjednik Povjerenstva:  
  
Prof. dr. sc. Branko Bauer

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
POPIS TABLICA.....	III
POPIS OZNAKA .....	IV
SAŽETAK.....	V
SUMMARY .....	VI
UVOD .....	1
1. PREGLED MPS-a .....	2
1.1 Funkcija cjelokupne stanice .....	2
1.2 Obradna stanica.....	3
2. MODERNIZACIJA STANICE ZA OBRADU.....	5
2.1. Zatečeno stanje.....	5
2.1.1 Upravljački modul.....	6
2.1.2 Dijagnostika robotske ruke .....	8
2.2 Programiranje u <i>Codesys</i> -u .....	10
2.2.1 Sekvencijski funkcijski dijagram (SFC) .....	10
2.2.2 Programiranje stanice za obradu .....	11
2.2.3 Download programa na PLC.....	14
3. PROGRAMIRANJE ROBOTA .....	16
3.1 Priprema robota.....	17
3.2 Programiranje robota.....	22
3.2.1 Programiranje u <i>COSIROP</i> -u.....	22
4. VIZUALIZACIJA U <i>CIROS</i> -u.....	27
4.1 Programski paket <i>CIROS</i> .....	27
4.2 Kreiranje vizualizacije .....	28
4.2.1 Modeliranje .....	28
4.2.2 Programiranje robota.....	32
ZAKLJUČAK .....	34
LITERATURA.....	35
PRILOZI.....	36
PRILOG A – Programski kod RV-M1 robota.....	37

**POPIS SLIKA**

Slika 1.	Podjela stanica [1] .....	2
Slika 2.	Primjer upravljačke ploče [1] .....	3
Slika 3.	Obradna stanica .....	4
Slika 4.	Zatečeno stanje MPS-a [1] .....	5
Slika 5.	Festo IPC [1] .....	6
Slika 6.	CECC-LK PLC [3] .....	7
Slika 7.	Sabirnica na stanici .....	7
Slika 8.	Prihvatnica [1] .....	8
Slika 9.	Mjerenje napona na prihvatnici [1] .....	8
Slika 10.	Lijevo ispravan tranzistor, desno neispravan [1] .....	9
Slika 11.	Primjer SFC jezika .....	10
Slika 12.	Programski dodatak .....	11
Slika 13.	Odabir jezika i modela PLC-a .....	12
Slika 14.	Sučelje <i>Codesys</i> -a .....	12
Slika 15.	Program stanice .....	13
Slika 16.	Pozivanje na funkcijski blok .....	14
Slika 17.	Funkcijski blok <i>Aktivacije</i> .....	14
Slika 18.	Kompiliranje koda .....	14
Slika 19.	<i>Festo Field Device Tool</i> .....	15
Slika 20.	<i>Communication settings</i> .....	15
Slika 21.	Mitsubishi Movemaster RV-M1 [6] .....	16
Slika 22.	Robotska stanica .....	17
Slika 23.	Prednja strana upravljačke jedinice [6] .....	18
Slika 24.	Stražnja strana upravljačke jedinice [6] .....	18
Slika 25.	Ubacivanje kartice u upravljačku jedinicu [6] .....	19
Slika 26.	Gornja strana upravljačke jedinice .....	19
Slika 27.	Sklope za definiranje rada robota .....	20
Slika 28.	Postavke DIP sklopki [6] .....	21
Slika 29.	Korisničko sučelje <i>COSIROP</i> -a .....	22
Slika 30.	Postavljanje komunikacije robota [10] .....	23
Slika 31.	Parametri serijske veze [9] .....	23
Slika 32.	Prozor za upravljanje robotom [9] .....	24
Slika 33.	Lista pozicija [9] .....	24
Slika 34.	Provjer sintakse programa [9] .....	25
Slika 35.	<i>Download</i> prozor [9] .....	25
Slika 36.	Pokretanje programa [9] .....	26
Slika 37.	Korisničko sučelje u <i>Ciros</i> -u [11] .....	27
Slika 38.	Dijelovi sučelja .....	28
Slika 39.	Postavljanje modela u radnom okruženju .....	29
Slika 40.	Svojstva modela .....	29
Slika 41.	Virtualna MPS stanica .....	31
Slika 42.	I/O ploča .....	31
Slika 43.	Otvaranje novog projekta .....	32
Slika 44.	Novi projekt .....	32
Slika 45.	Programiranje robota .....	33

---

**POPIS TABLICA**

Tablica 1. Osnovni tehnički podaci za robota [6] ..... 16  
Tablica 2. Hijerarhija tipova elemenata u modeliranju ..... 30



---

**POPIS OZNAKA**

<b>Oznaka</b>	<b>Opis</b>
FBD	Function block diagram
FER	Fakultet elektrotehnike i računalstva
IL	Instruction list
IP	Internet protocol
IPC	Industrial PC
LD	Ladder diagram
MPS	Modular production system
POU	Program Organisation Unit
PLC	Programmable logic controller
SFC	Sequential function chart
ST	Structured text
TŠRB	Tehnička škola Ruđera Boškovića

---

**SAŽETAK**

Modularni proizvodni sustav je didaktička radna stanica za učenje i razvijanje vještina potrebnih u proizvodnji. Njena svrha je da na jednostavan način omogući korisniku da složi model svoje proizvodne linije.

U ovom završnom radu se postojeći modularni proizvodni sustav tvrtke Festo koji se sastoji od pet radnih stanica dokumentirao i modernizirao suvremenim programabilnim logičkim kontrolerima (PLC).

Cilj ovog rada bio je opisati modularni proizvodni sustav, programirati radne stanice pripadajućim PLC-ima, izraditi pneumatske sheme te ovladati i primijeniti novije programske alate za automatizaciju poput *Codesys*-a i *Ciros*-a.

Ključne riječi: modularni proizvodni sustav, programiranje, modernizacija, pneumatika

---

**SUMMARY**

The modular production system is a didactic workstation for learning and developing skill required in production. Its purpose is to easily allow the user to design a modular production line. In this final work, the existing modular production system system of Festo, consisting of five workstations, has been documented and modernized by modern programmable logic controllers (PLC).

The aim of this paper was to describe the modular production system, program the workstations with associated PLCs, create pneumatic schemes, and master and apply newer automation tools such as *Codesys* and *Ciros*.

Key words: modular production system, programing, modernization, pneumatics

## UVOD

Napretkom tehnologije i povećanjem njene kompleksnosti sve je veća potreba za mladim i vještim radnicima, koji su već imali iskustva u praksi. No praksa u stvarnim pogonima često nije moguća te je mladim radnicima teško steći kvalitetno radno iskustvo. Jedno od mogućih rješenja jest simuliranje velikih proizvodnih pogona korištenjem osnovnih funkcija proizvodnje kao vodiljama za izradu manjih modularnih proizvodnih stanica.

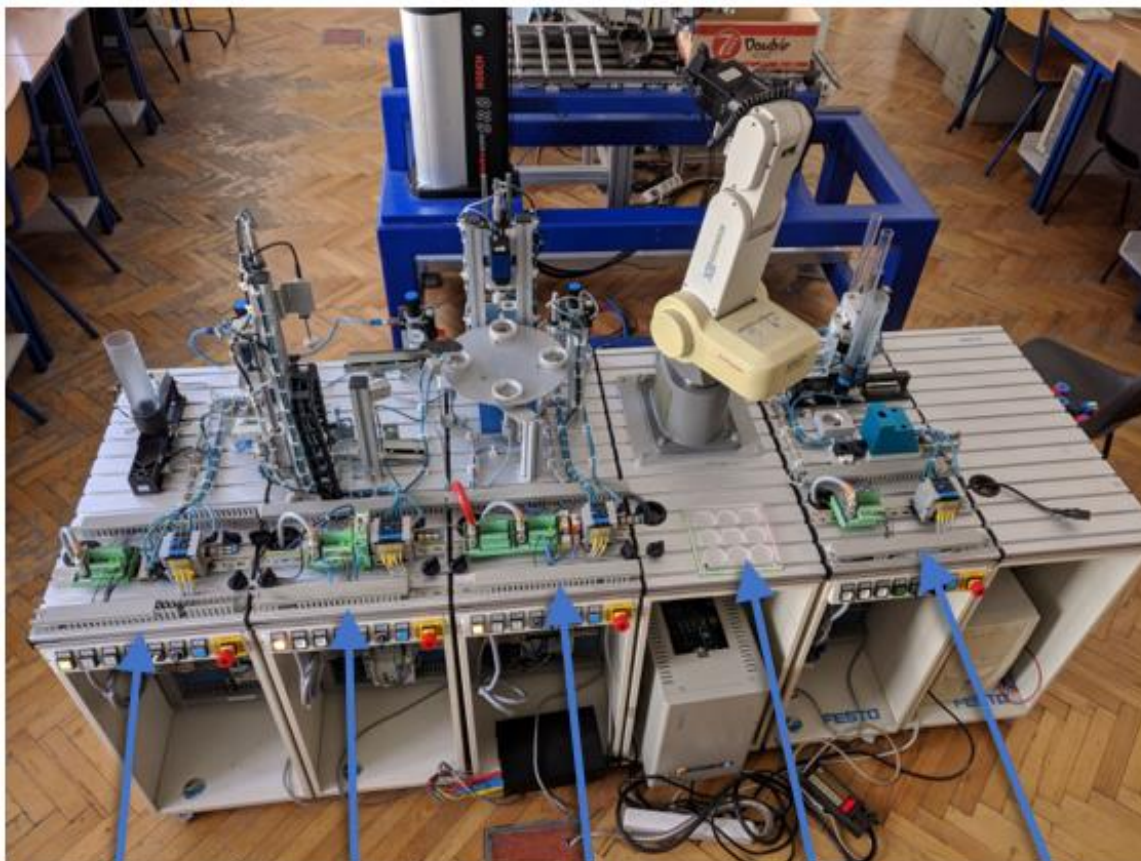
Modularni proizvodni sustav (engl.. *Modular Production System – MPS*) je didaktička radna stanica za učenje i razvijanje vještina potrebnih u proizvodnji. Njena svrha je da na jednostavan modularan način omogući korisniku da složi svoju proizvodnu liniju. Svaka linija se sastoji od modula koju su reprezentirani raznim elektroničkim i pneumatskim elementima postavljenim na metalna kolica koja se mogu pomicati i slagati jedna do druge u razne kombinacije. Također, moguće je pomicati i razmještati same elemente po stanicama. Svaki modul predstavlja neku funkciju u stvarnoj proizvodnji te tako imamo funkcije poput distribucije, mjerenja, pakiranja, obrade, sortiranja, robotske ruke i drugih. Ovaj završni rad napravljen je djelomično u suradnji s kolegom studentom Lukom Vučetićem, koji je svoj dio rada opisao u [1]. Stoga su i poglavlja koja opisuju MPS (Poglavlje 1) i početno stanje MPS-a (Poglavlje 2) zajednički napisana te se u velikoj mjeri podudaraju.

Stanica na kojoj je rađeno koristi pneumatske i električne komponente njemačke firme Festo te robota japanske tvrtke Mitsubishi. Festo Hrvatska osim svoj prodajnog ureda ima i edukacijski laboratorij unutar Tehničke škole Ruđera Boškovića (TŠRB) ) u Zagrebu koji je opremljen suvremenom Festo-vom opremom u kojem se nalazi i modularna proizvodna stanica iz 90-ih godina prošlog stoljeća. U ovom završnom radu je zadatak bio modularnu proizvodnu stanicu modernizirati suvremenim programabilnim logičkim kontrolerima (engl.. *Programmable logic controller – PLC*) i izraditi potrebnu dokumentaciju pneumatskih shema. Također je bitan aspekt ovog rada bio ovladati i primijeniti novije programske alate za automatizaciju poput *Codesys-a* i *Ciros-a*.

## 1. PREGLED MPS-a

### 1.1 Funkcija cjelokupne stanice

Svrha MPS-a je raznobojne obratke izrađene od metala i plastike razvrstati, te prihvatljive izratke testirati, obraditi ih i pomoću robota sklopiti. Proces se odvija preko pet modularnih koje su redom: distribucijska stanica, ispitna stanica, obradna stanica, robotska stanica te sklopna stanica. U ovom završnom radu opisane su obradna i robotska stanica, dok se distribucijska i ispitna stanica obrađena u završnom radu Luke Vučetića [1]. Svaka stanica koristi stlačeni zrak, radnog tlaka 6 bara, kao medij pogona aktuatora i električni signal od 24V za upravljanje.



Distribucijska  
stanica

Ispitna  
stanica

Obradna  
stanica

Robotska  
stanica

Sklopna  
stanica

Slika 1. Podjela stanica [1]

Svaka stanica, osim robotske, ima vlastitu upravljačku ploču na kojoj se nalaze elektroničke komponente poput tipkala, sigurnosnog releja, sabirnice, razdjelnika napona, osigurača i industrijskog računala (engl. *Industrial PC – IPC*).



Slika 2. Primjer upravljačke ploče [1]

## 1.2 Obradna stanica

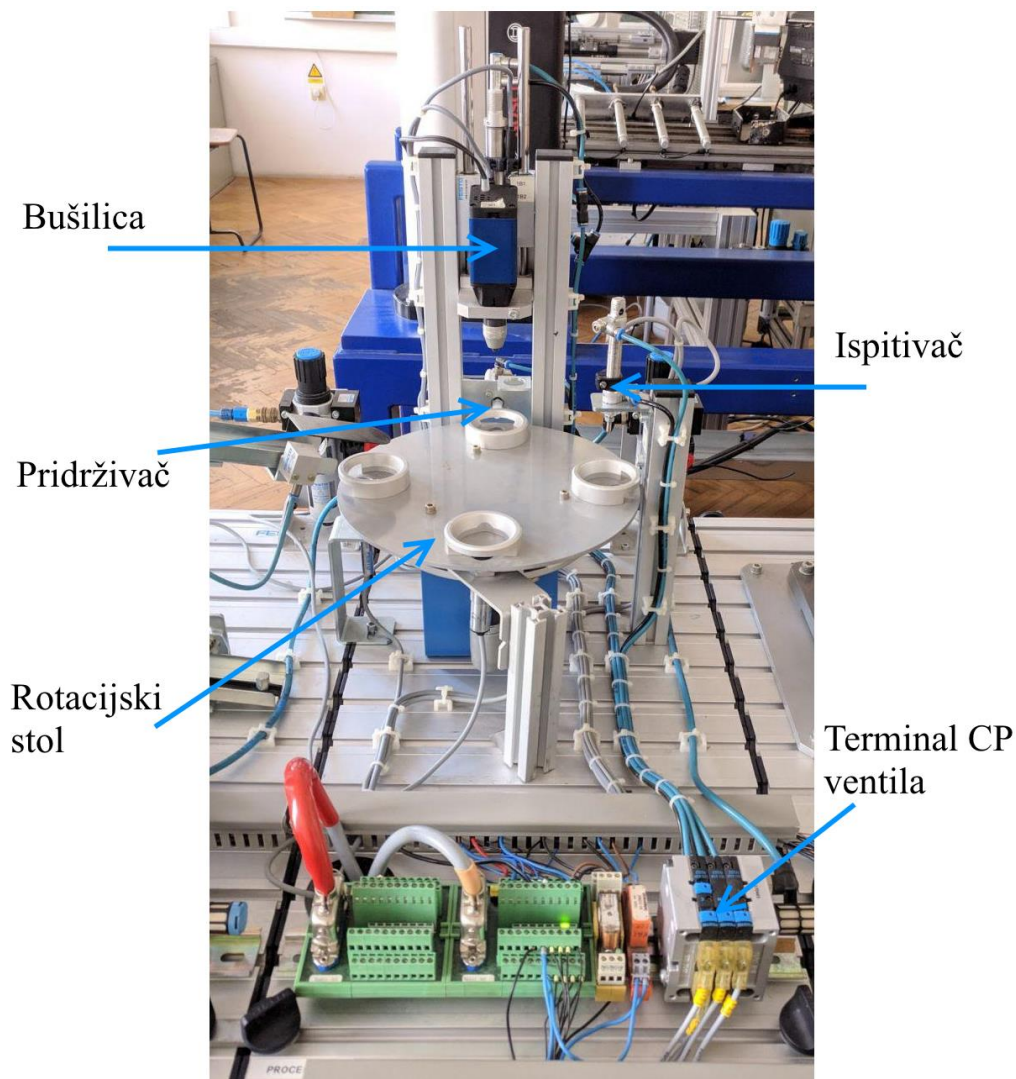
Prema [2], obrada je generički izraz za proizvodne korake kao što su formiranje, promjena oblika, strojna obrada i spajanje. Formiranje je stvaranje geometrijski determiniranog oblika iz bezoblične mase. Promjena oblika predstavlja promjenu geometrijskih oblika i/ili dimenzija tijela. Strojna obrada predstavlja promjenu materijalnih svojstava i/ili površina tijela. Spajanje označuje trajni spoj više tijela.

Funkcije obradne stanice su:

- obraditi komad
- provjeriti ispravnost rupe
- pripremiti komad sljedećoj stanici

U stanici za obradu, obradak se ispituje i obrađuje na rotacijskom stolu. Rotacijski stol za indeksiranje pokreće DC motor. Stol se pozicionira pomoću relejnog kruga, a položaj stola detektira induktivni senzor. Završeni obradak se pomoću robota prenosi dalje. Na slici 3. prikazani su osnovni elementi obradne stanice.

Nakon što je obradak, preko tobogana, stigao u prvi položaj, kreće rotacija stola za 90°. Prvo se aktivira pridržiivač predmeta kod bušilice. Poslije toga pali se bušilica, pomiče prema dolje, zadržaje par sekundi i opet vraća u prvotni položaj. Stol se neće nastaviti okretati dok ne dođe novi obradak u prvi položaj. Kada stigne novi obradak, onaj prvi dolazi u treći položaj rotacijskog stola. Spušta se ispitivač te ustvrđuje kvalitetu rupe. Nakon toga stanica opet čeka da stigne novi obradak u prvi položaj. Kad se to dogodi, onaj prvi obradak dolazi u posljedni položaj, a PLC daje signal robotu da preuzme obradak. Proces se neće nastaviti dok zadnji položaj na stolu ne bude prazan.

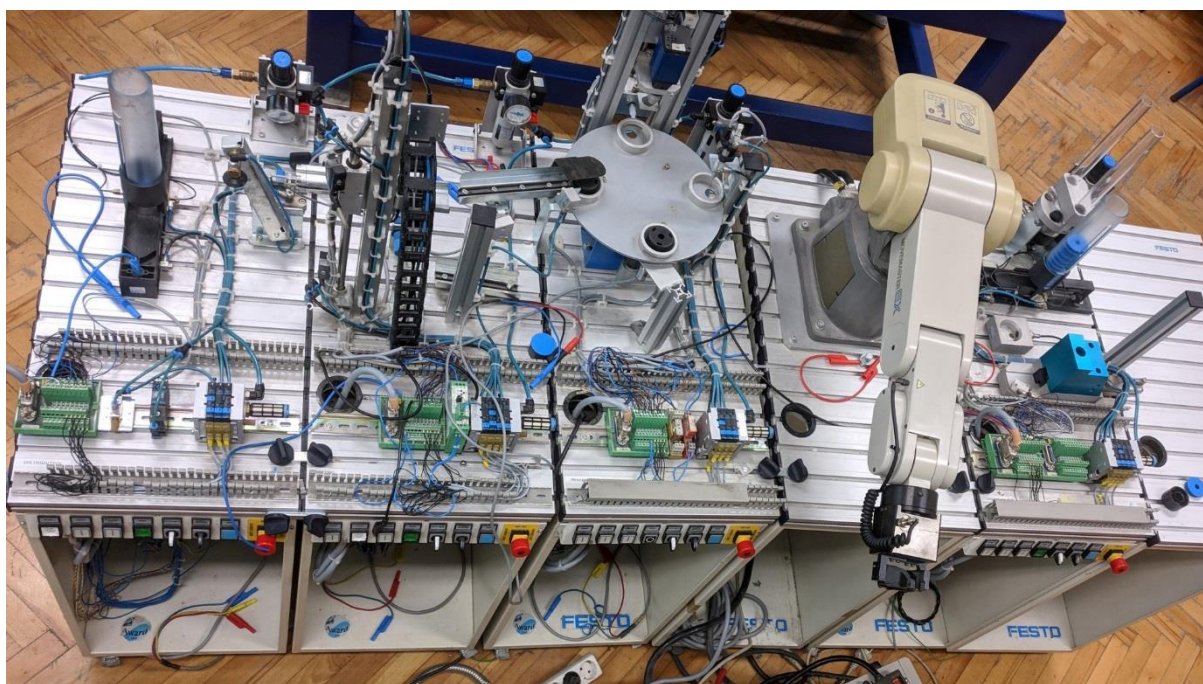


Slika 3. Obradna stanica

## 2. MODERNIZACIJA STANICE ZA OBRADU

### 2.1. Zatečeno stanje

Modularni proizvodni sustavi koriste se u edukacijske svrhe već dugi niz godina. MPS na kojem je rađeno kupljen je 2001. godine i u upotrebi je bio na Fakultetu elektrotehnike i računarstva (FER), gdje se tada nalazio Festo-v laboratorij, do 2013. godine kada je premješten u Tehničku školu Ruđera Boškovića (TŠRB). Otkako je postavljen u učionici robotike TŠRB-a, MPS nije bio u funkciji. Prije početka rada, na njemu se provela dijagnoza stanice te je ustanovljeno da je potrebno zamijeniti zastarjela industrijska računala (engl. Industrial PC – IPC) suvremenim Festo-vim PLC-ima te da je robotska ruka u kvaru. Također, bilo je potrebno zamijeniti oštećenu vakuumsku hvataljku novom, manjeg promjera, popraviti dotrajala ožičenja, projektirati dovod stlačenog zraka stanici kompresorom i napajanja PLC-a.



Slika 4. Zatečeno stanje MPS-a [1]



### 2.1.1 Upravljački modul

Za upravljanje stanicama korišteni su Festo-vi IPC-i koji se sastoje od baze i modula. Usprkos velikoj popularnosti industrijskih računala, brzo su ih zamijenili PLC-i koji su donijeli napredak u vidu jednostavnijeg programiranja.



Slika 5. Festo IPC [1]

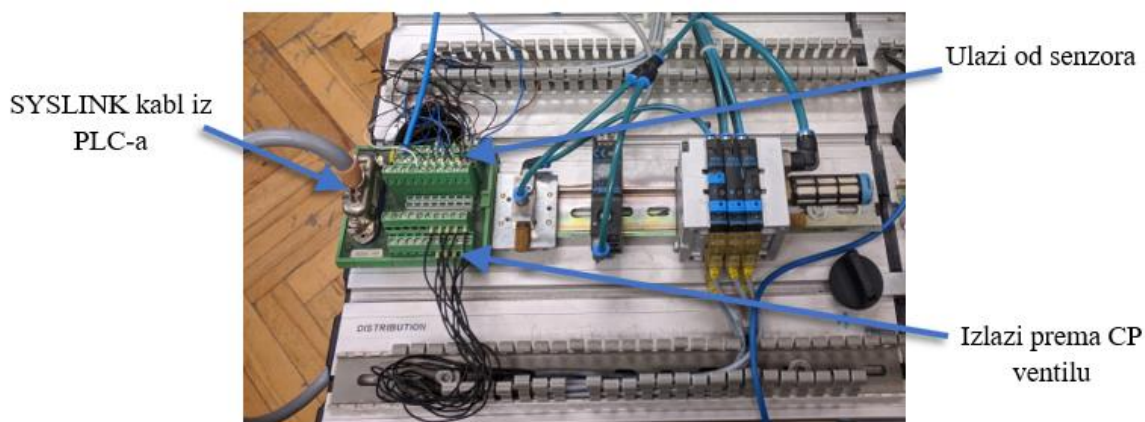
Festo je omogućio korištenje njihovih novih modela PLC-a kao zamjenu za stare IPC-e. Korištena su dva CECC-LK kontrolera i jedan CPX-E kontroler, koji su raspoređeni na distribucijsku stanicu, ispitnu stanicu te obradnu stanicu.

CECC-LK kontroler nije modularan te ima 14 digitalnih ulaza i 8 izlaza, a komunikacija s računalom se vrši mrežnim kablom preko rutera. Jedan kontroler je postavljen na distribucijsku stanicu, a drugi na obradnu stanicu.



Slika 6. CECC-LK PLC [3]

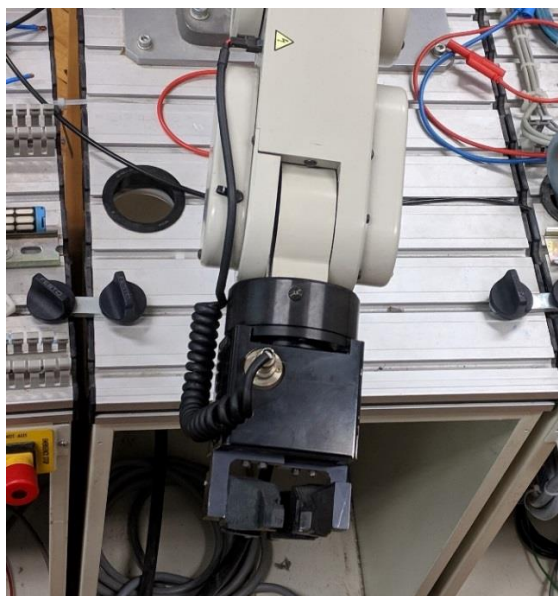
Radi nedostatka dokumentacije stanice bilo je potrebno eksperimentalno, modelom crne kutije, saznati funkcije svake stanice i njihovo ožičenje. Ispitivanja smo proveli dovodeći vanjski napon od 24V na izlaze IPC-a čime smo dobili jasan uvid u funkciju svakog voda. Izlazi iz IPC-a nisu spojeni direktno na terminal CP ventila, već su dio SYSLINK kabla koji prenosi signale i napon na sabirnicu postavljenu na samu stanicu. Time se omogućuje jednostavno i brzo rastavljanje upravljačke ploče od same stanice te postiže organiziraniji i uredniji radni prostor.



Slika 7. Sabirnica na stanici

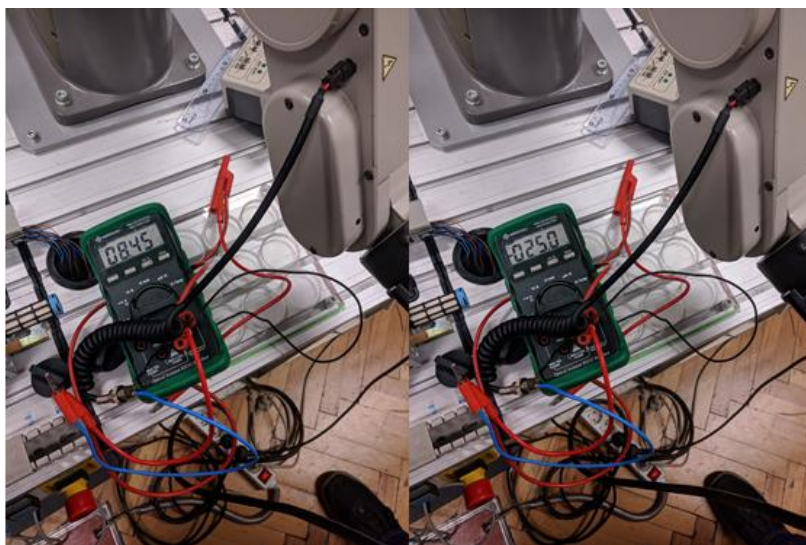
### 2.1.2 Dijagnostika robotske ruke

Važan element ovog MPS-a je robotska ruka, koja se nalazi na četvrtoj stanici u redu. Robot nije bio u upotrebi već dugi niz godina te je bilo potrebno uspostaviti vezu s računalom preko RS232 kabla i namjestiti valjane postavke sklopki na upravljačkoj jedinici. Opširniji opis se nalazi u poglavlju 3. Slijedilo je testiranje robota. Nakon opsežnog testiranja zaključeno je da su svi zglobovi robota funkcionalni osim signala otvaranja i zatvaranja prihvatnice.



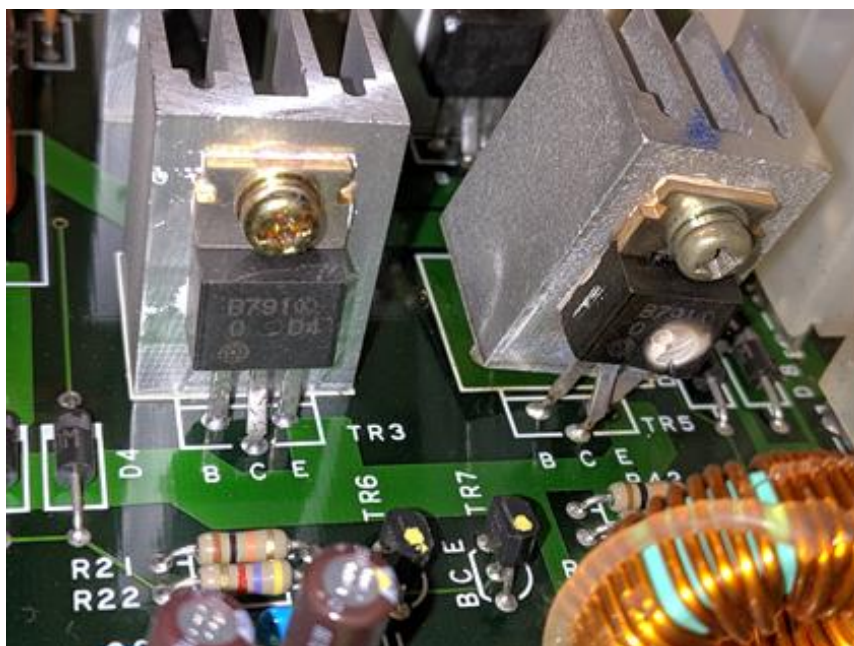
Slika 8. Prihvatnica [1]

Pronalazak kvara započeo je sistematski od same prihvatnice. Prihvatnicom se upravlja signalom od  $\pm 24V$  te je prvi korak bio provjeriti napon na prihvatnici.



Slika 9. Mjerenje napona na prihvatnici [1]

Mjerenjem je ustanovljeno da stvarni napon ne odgovara potrebnom upravljačkom naponu od  $\pm 24V$ , nego iznosi  $+ 8V$  i  $- 25V$ . Izmjerene vrijednosti nam sugeriraju da je problem u upravljačkoj jedinici. Nabavkom servisne dokumentacije od Mitsubishi-ja (zahvaljujući kolegi Dominiku Vezmaru iz Inea CR d.o.o.) dolazi se do zaključka da je problem u H-mostu koji regulira polaritet napona prihvatnice te omogućuje njeno otvaranje/zatvaranje. Rastavljanjem upravljačke jedinice robota pronađen je pregorjeli tranzistor spomenutog H-mosta.



**Slika 10. Lijevo ispravan tranzistor, desno neispravan [1]**

Popravak robotske ruke ne ulazi pod zadatak ovog završnog rada te je nakon uspješne dijagnostike kvara, rad na robotu završen.

## 2.2 Programiranje u Codesys-u

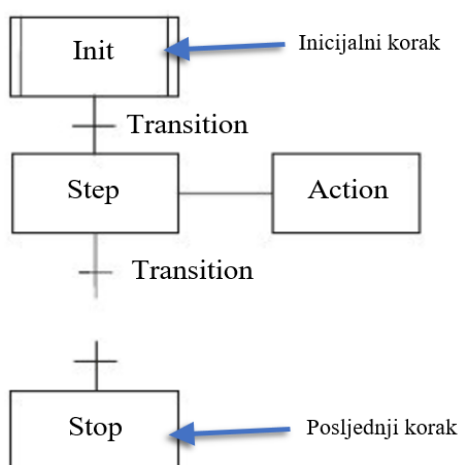
Standardizacija programiranja upravljačke logike donijela je nebrojene prednosti u interoperabilnosti proizvoda različitih proizvođača. Međunarodni standard koji pokriva, između ostaloga, programske jezike za PLC-ove naziva se IEC 61131-3 [7] i on definira tri grafička i dva tekstualna jezika:

- Ladder dijagram (LD), grafički
- Funkcijski blok dijagram (FBD), grafički
- Dijagram slijednih funkcija (eng. *Sequential function chart*, (SFC)), grafički
- Strukturirani tekst (ST), tekstualni
- Popis instrukcija (eng. *Instruction list*, (IL)), tekstualni jezik

*Codesys* (skraćenica od „**C**ontroller **d**evelopment **s**ystem“) je besplatno razvojno okruženje za programiranje PLC-ova u skladu sa IEC 61131-3 standardom. Podržava programiranje u svih pet jezika definiranih standardom [5]. U ovom radu za programiranje stanice koristi se programski paket *Codesys v3.5*.

### 2.2.1 Sekvencijski funkcijski dijagram (SFC)

Za programiranje stanica korišten je SFC grafički jezik, koji je prilagođen za programiranje sekvencijskih automata, baziran na Petri mrežama i Grafsetu te podržava više istovremenih stanja. Graf se sastoji od linija toka (engl. *flowlines*), prelazaka (engl. *transition*), koraka (engl. *step*) i akcija (engl. *action*) [4].



Slika 11. Primjer SFC jezika

Sam kod funkcionira na način da postoje prelasci (Transition) koji moraju biti zadovoljeni kako bi tok prešao na idući korak (Step). Tok se zadržava na koraku sve dok se idući uvjet ne ispuni, nakon čega tok prelazi na idući korak i tako redom dok program ne dođe do kraja ili ne postoji petlja koja ga vraća ponovo u kod. Svakom koraku se mogu dodijeliti ulazne, trenutne i izlazne akcije (Action). Ulazne akcije vrše svoju radnju prilikom dolaska toka na korak, trenutne akcije vrše svoju radnju dok je god korak aktivan, dok izlazne akcije vrše svoju radnju prilikom izlaska toka iz tog koraka.

Velika prednost SFC jezika nad drugim spomenutim jezicima je upravo njegova jednostavnost, vrlo ga je lagano čitati i otkloniti probleme u kodu, što omogućuje i manje iskusnim radnicima da ga razumiju i održavaju.

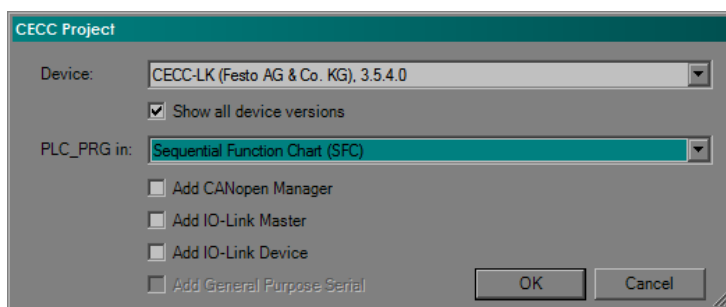
## 2.2.2 Programiranje stanice za obradu

Programski paket *Codesys* moguće je preuzeti besplatno s Festo web-stranice. Osim programa, nužno je preuzeti dodatne *software*-ske pakete za PLC koji koristimo i instalirati ih kao priključak *Codesys*-u. Odraban je dodatak za Festo-ov CECC-LK PLC pošto se on nalazi na stanici. Na slici 12. je prikazan iskječak s službene Festo web-stranice odakle je preuzet taj dodatak [8].



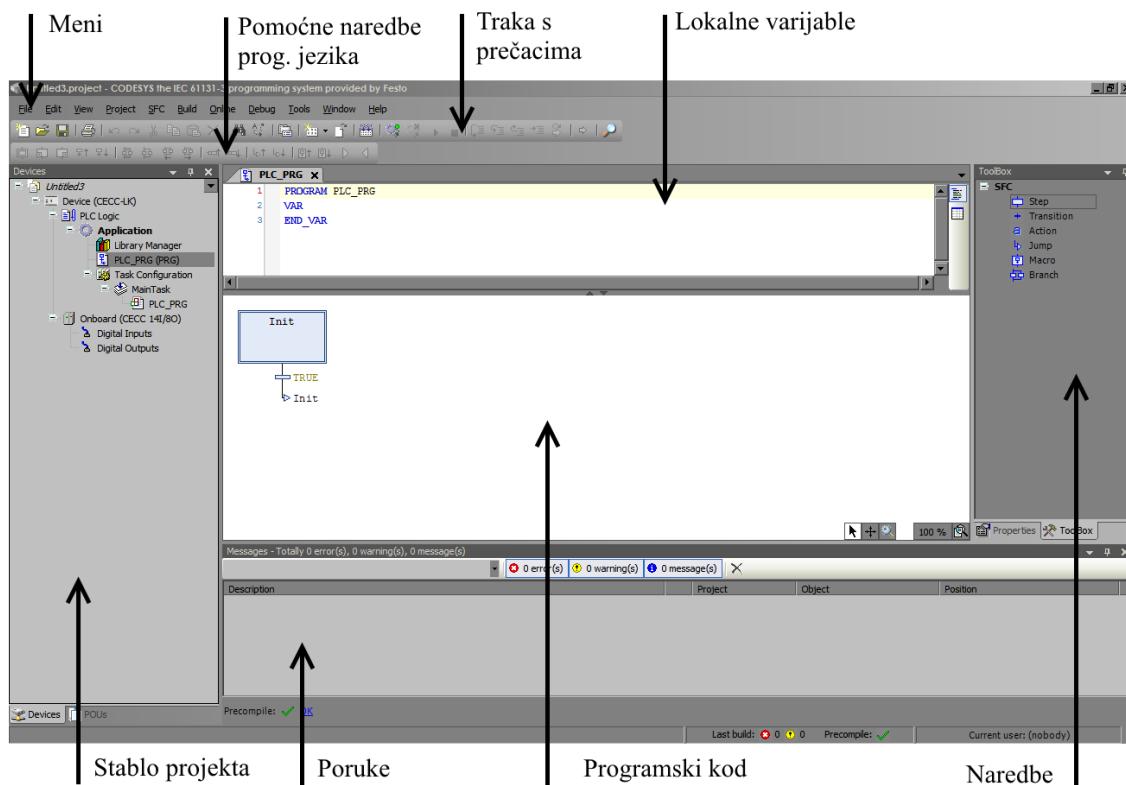
Slika 12. Programski dodatak

Nakon uspješne instalacije, potrebno je otvoriti novi projekt unutar programa te definirati jezik u kojem će glavni program biti napisan. U ovom radu korišten je Dijagram sljednih funkcija (eng. *Sequential function chart*). Također, u istom prozoru gdje biramo jezik potrebno je odabrati ispravan model PLC-a (slika 13.).



Slika 13. Odabir jezika i modela PLC-a

Klikom na „OK“ dočekać će nas sučelje kao na slici 14. Na istoj slici su, radi lakšeg razumijevanja daljnjeg teksta, napisana imena pojedinih dijelova sučelja.



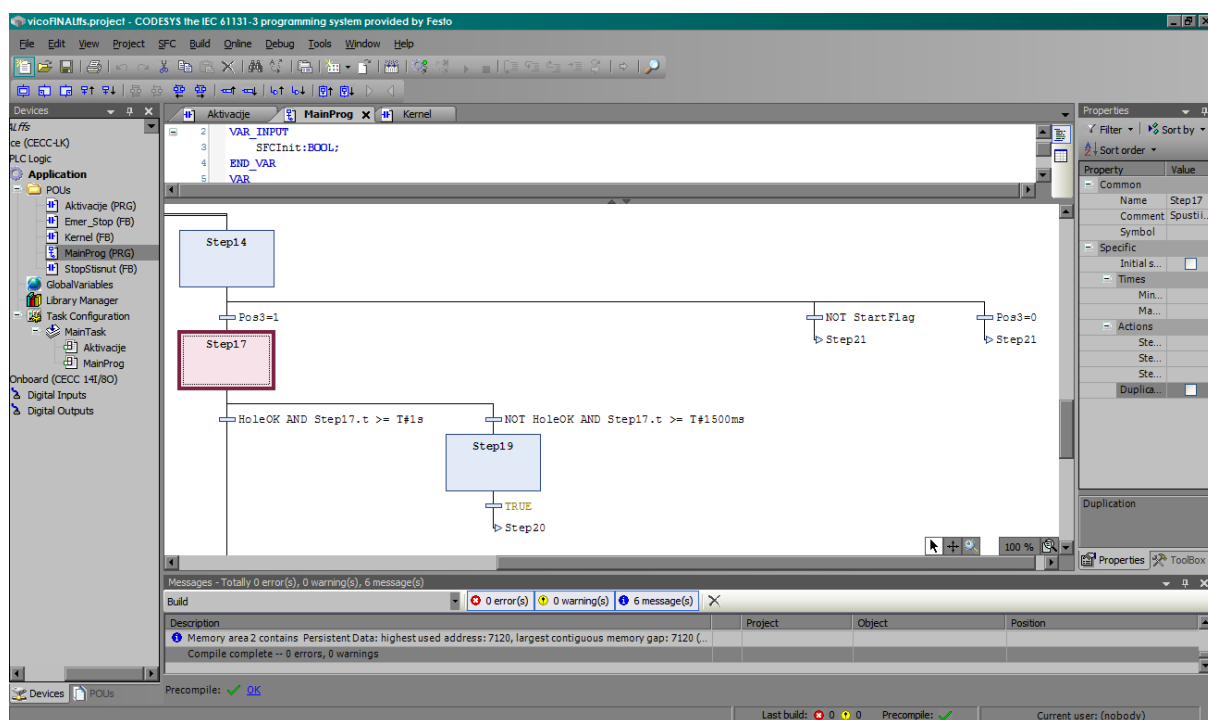
Slika 14. Sučelje Codesys-a

Na slici 14. s lijeve je strane vidljivo stablo projekta na kojem radimo i ono se sastoji od:

- **Application** – program na PLC-u koji se sastoji od:
  - **Library manager** – kolekcija funkcija za neki *hardware*

- **POU (Program Organisation Unit)** – dio cjelokupnog programa koji izvršava neki zadatak. Dalje se može podijeliti na: *Program*, *Function Block* i *Function*. *Program* definira programski kod kojeg PLC izvršava. *Function block* predstavlja programski kod koji se poziva unutar *Program*-a. Primjer jednog *Function*-a je matematička funkcija sinus, a možemo definirati sami svoju funkciju ili koristiti gotovu iz *Library manager*-a.
- **Global variables** – varijable kojima se može koristiti iz bilo kojeg POU-a
- **Task configurator** – U *Task configurator*-u dodajemo svoje POU-ove i pridodjeljujemo im prioritet po kojem će se izvršavati.
- **Onboard Inputs/Outputs** – pregled stanja I/O na PLC-u

Na slici 15. je prikazan isječak programa za obradnu stanicu.

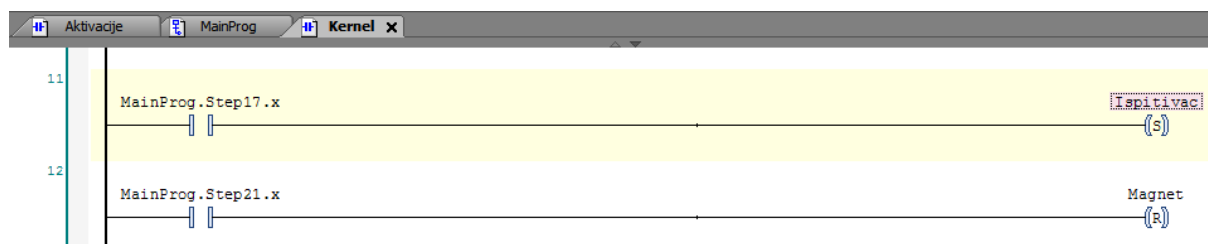


Slika 15. Program stanice

U stablu projekta pod *Task Configuration* vidljivo je da postoje dva programa koji se kontinuirano izvršavaju u pozadini. Program/POU *MainProg* predstavlja „kostur“ cijele stanice i o njemu najviše ovisi kako će stanica raditi. Svaki program/POU ima mogućnost da se poziva dijelove koda nekog funkcijski blok. Npr. na slici 15. je prikazan dio koda koji upravlja ispitivačem rupe na obradcima. Kada program dođe do koraka „Step17“, istovremeno se

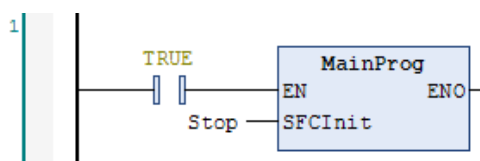


aktivira određena grana u Ladder dijagramu funkcijskog bloka „Kernel“ (slika 16.). Time se izlaz „Ispitivac“ postavlja u stanje „Set“.



Slika 16. Pozivanje na funkcijski blok

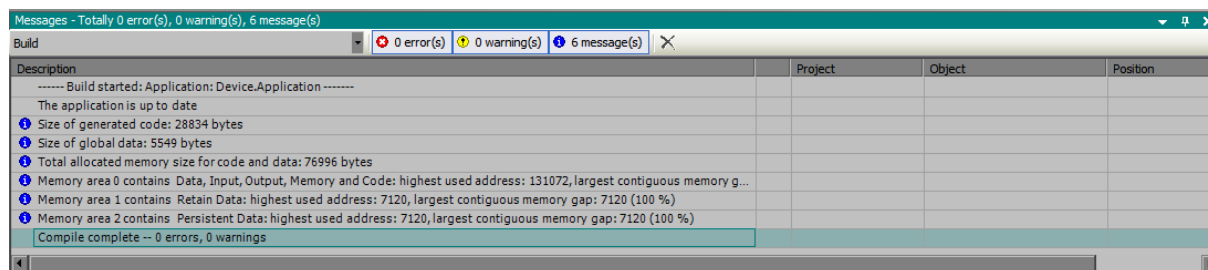
Svrha programa/POU-a *Aktivacije* je, između ostaloga, da kontinuirano provjerava da li je stisnuta tipka „Stop“ na obradnoj stanici. Kako je to bitna zadaća gledajući s aspekta sigurnosti, taj program dobiva veći prioritet od programa *MainProg* (slika 15.). Na slici 17. je prikazan funkcijski blok koji izvršava taj zadatak. Kada se stisne tipka „Stop“, program *MainProg* zaustavlja rad dok se ne stisne tipka „Reset“.



Slika 17. Funkcijski blok *Aktivacije*

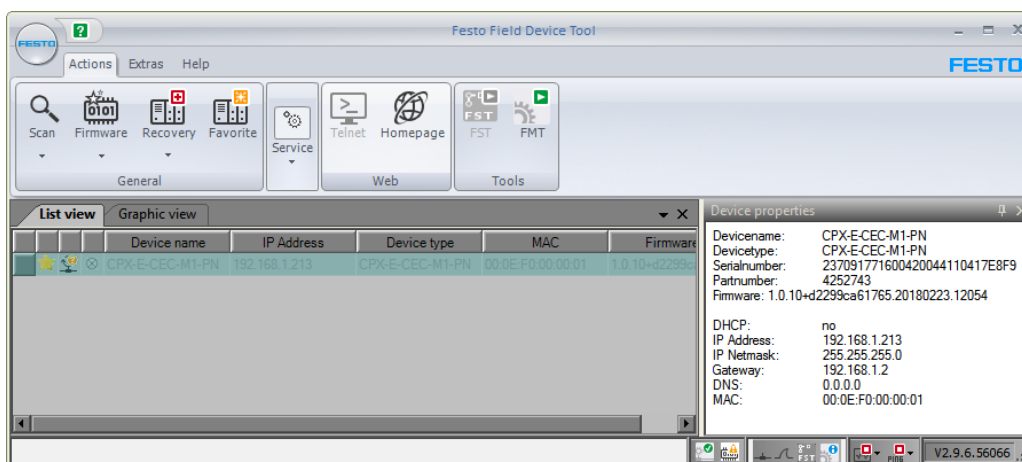
### 2.2.3 Download programa na PLC

*Download*-u cjelokupnog programa prethodi uspješnom kompiliranju koda. Ako kompiliranje uspješno završi, dobije se poruka kao na slici 18.

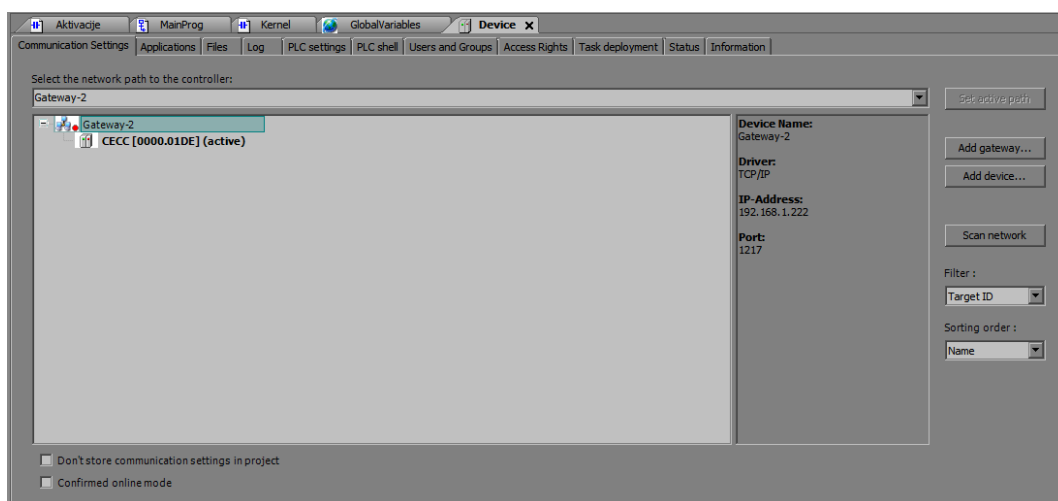


Slika 18. Kompiliranje koda

U ovom radu je komunikacija između PLC-a i računala uspostavljena preko Ethernet kabela. Oba uređaja je bilo potrebno spojiti na istu lokalnu mrežu (npr. router), a potom identificirati IP adresu PLC na mreži. U tome je pomogao alat *Festo Field Device Tool*. Nakon pokretanja programa, potrebno je pokrenuti *Scan* te će se kreirati lista svih PLC-ova na mreži (slika 19.).

Slika 19. *Festo Field Device Tool*

Nakon što smo pronašli IP adresu PLC-a, potrebno je unutar *Codesys*-a definirati *Gateway* odnosno uspostaviti komunikaciju. Dvostrukim klikom na ime uređaja u stablu projekta otvara nam se *Communication settings* gdje je potrebno kliknuti na *Scan network*. Nakon kraćeg vremena uređaj će biti pronađen i postati aktivan (slika 20.).

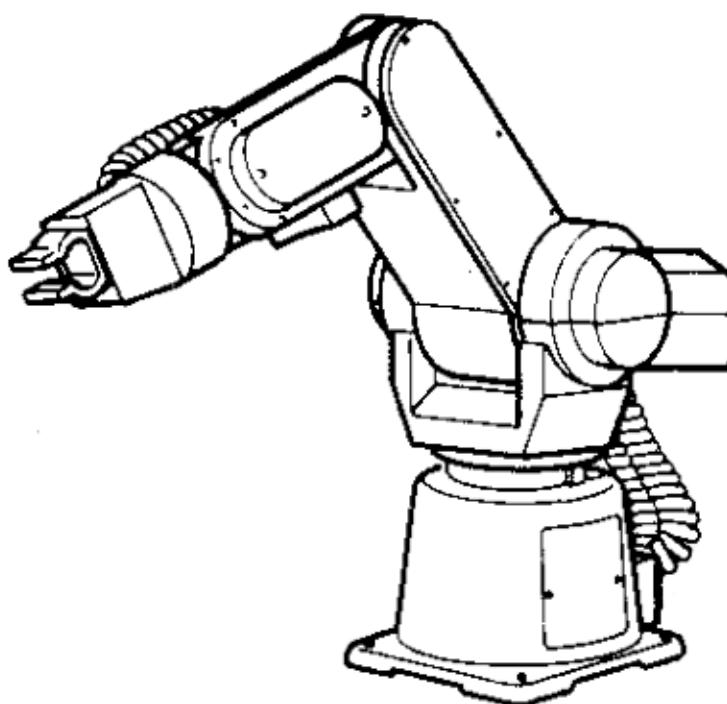
Slika 20. *Communication settings*

Sada je moguće unijeti projekt na PLC tako da se klikne na dugme *Login* u traci s prečacima ili tako da koristimo „Alt+F8“. U donjem dijelu ekrana pisati će „Stop“ što označava da aplikacija nije pokrenuta. Pritiskom na „F5“ ona se pokreće. Dok je aplikacija pokrenuta moguće je *real-time* pratiti izvršavanje koda, stanje I/O, vrijednosti varijabli itd. Da bi aplikacija ostala na PLC-u i nakon otpovezivanja s računalom, potrebno je samo kliknuti na *Create boot application* pod *Menu > Online*.

### 3. PROGRAMIRANJE ROBOTA

Da cijeli proces ne bi završio na prethodno opisanoj stanici za obradu, potrebno je uvesti robota koji bi obratke prenosio dalje. Ova MPS stanica koristi Mitsubishi robot, model RV-M1 (Movemaster EX).

Robot posjeduje 5 stupnjeva slobode sa nosivošću od 1,2 kg (bez hvataljke). Unatoč maloj veličini, robot se koristi u industrijske svrhe [6]. Programiranje robota moguće je izvesti preko privjeska za učenje (eng. *Teaching box*) ili preko osobnog računala. Ostali tehnički podaci za navedenog robota se nalaze u tablici 1.

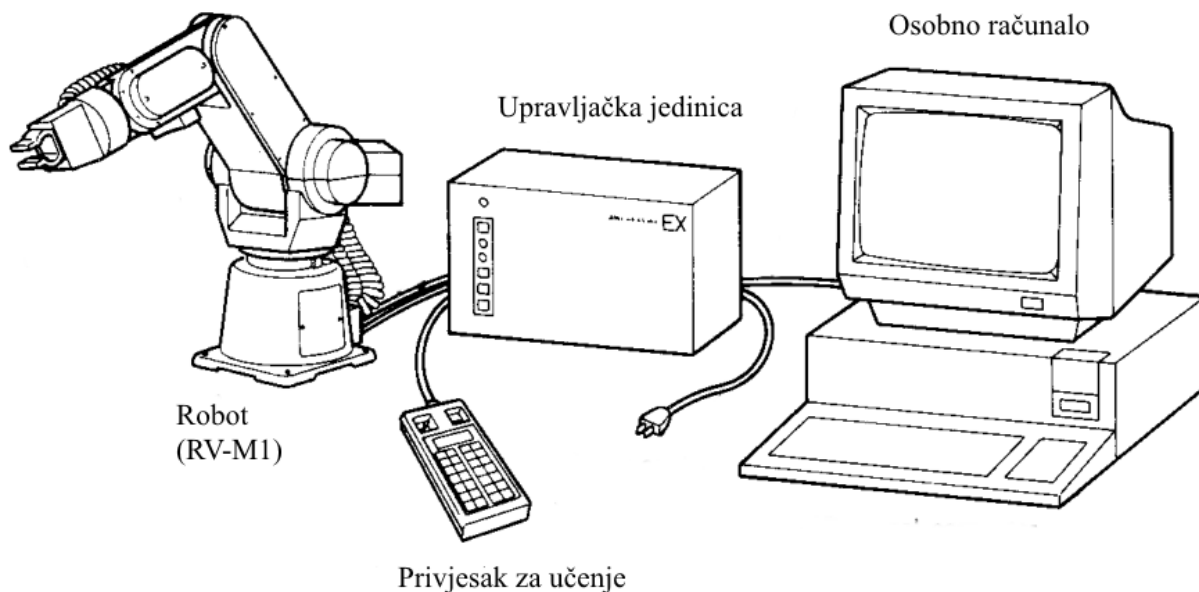


Slika 21. Mitsubishi Movemaster RV-M1 [6]

Tablica 1. Osnovni tehnički podaci za robota [6]

Veličina	380 (širina) x 331 (dužina) x 246 (visina) mm
Težina	cca. 23 kg
Napajanje	120V/220V/230V/240V
Brzina	1000mm/s (10 stupnjeva br.)
Pohrana	EP-ROM, baterijski napajan RAM
Maks. br. linija koda	2048 (16 KB)
Maks. br. pozicija	629 (8 KB)
Pogon hvataljke	DC ili pneumatski (preko AC solenoida)

### 3.1 Priprema robota



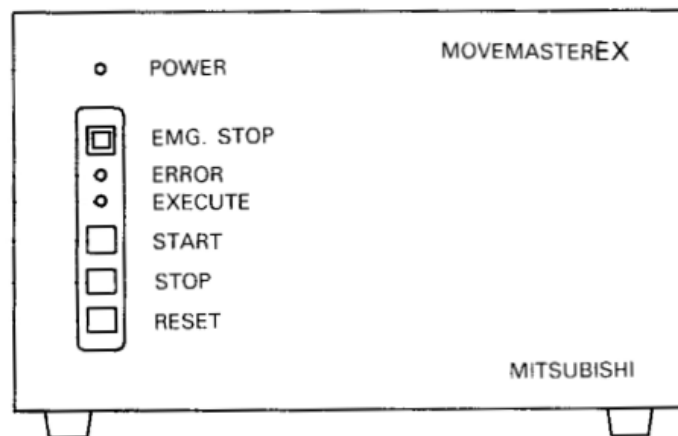
Slika 22. Robotska stanica

Postoje dva načina na koji se može konfigurirati rad *Movemaster* robotskog sustava:

- **Sustav centriran oko računala** – kod ovog načina rada računalo služi kao „mozak“ cijele operacije. Pogodan za sustave gdje računalo treba obraditi informacije iz različitih izvora (kamere, mjerači i sl.).
- **Sustav centriran oko upravljačke jedinice** – u ovom načinu rada računalo služi samo za programiranje robota. Robot postaje neovisan o računalu, jer se program sprema u ugrađenu EP-ROM memoriju. Izmjena signala sa drugim uređajima se odvija preko I/O priključka u upravljačkoj jedinici.

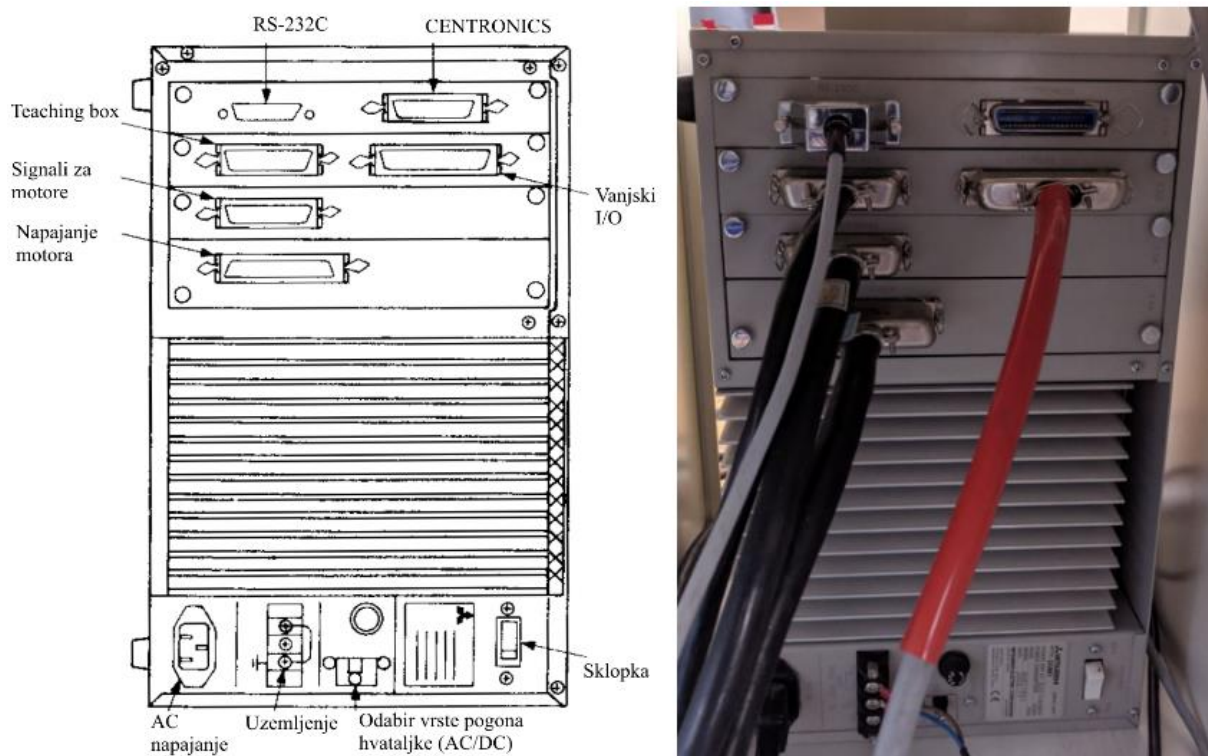
#### 3.1.1 Konfiguriranje upravljačke jedinice

Na prednjoj strani upravljačke jedinice nalaze se tipkala za upravljanjem programom, *emergency stop* tipkalo i tri LED indikatora (slika 23.). Prvi LED indikator označava da li je uređaj uključen. Drugi LED indikator (na slici nazvan „ERROR“) upozorava da se dogodila pogreška u sustavu. Indikator može javljati različite vrste pogreški ovisno o tome koliko brzo LED-ica treperi. Naposljetku, treći LED indikator označava da se izvršava program ili komanda zadana privjeskom za učenje.



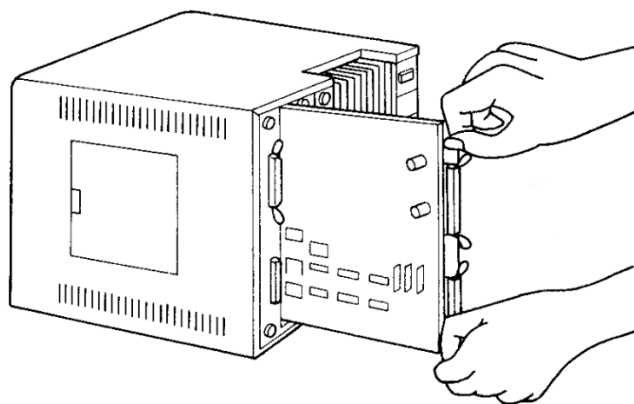
Slika 23. Prednja strana upravljačke jedinice [6]

S stražnje strane upravljačke jedinice nalaze se konektori, napajanje, hladnjak, sklopka za paljenje i druga sklopka za odabir vrste pogona hvataljke (slika 24.). Na ovoj stanici sklopka za pogon hvataljke je postavljena u poziciju „DC“ pošto koristimo električnu hvataljku. U slučaju da koristimo pneumatsku hvataljku sklopka se mora postaviti u poziciju „AC“.



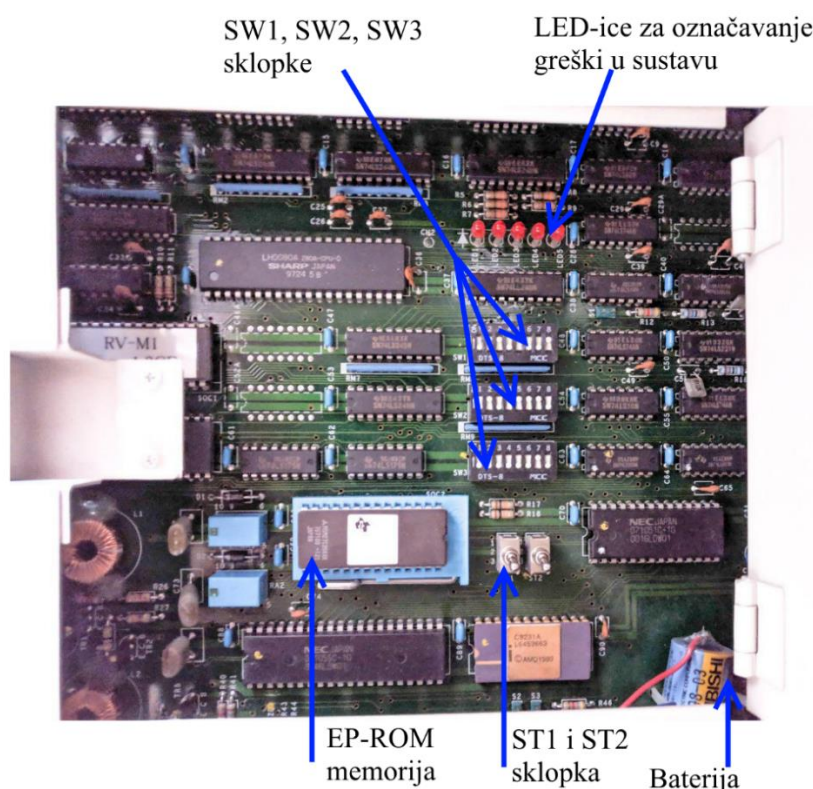
Slika 24. Stražnja strana upravljačke jedinice [6]

Konektori na *Movemaster* robotu su dio kartica koje se ubacuju u kućište upravljačke jedinice na način kao što je prikazan na slici 25. Karticu za I/O je moguće zamjeniti s boljom varijantom koja ima 16 ulaza i izlaza.



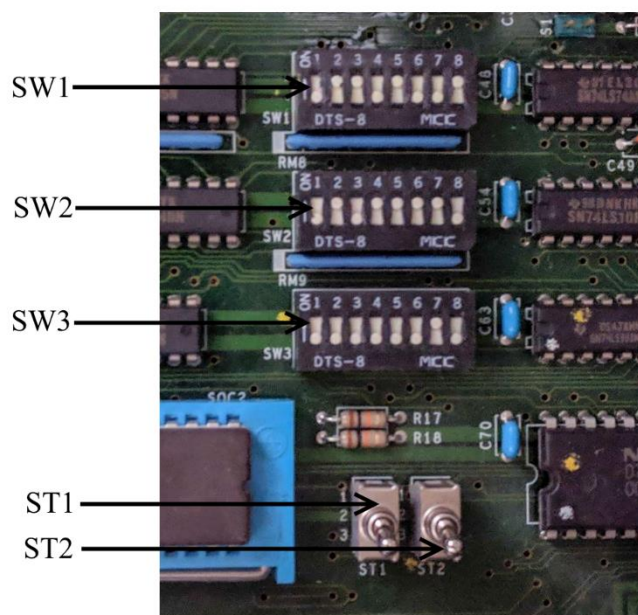
**Slika 25. Ubacivanje kartice u upravljačku jedinicu [6]**

S gornje strane kućišta upravljačke jedinice nalazi se poklopac koji omogućuje pristup najgornjoj kartici, a to je kartica za komunikaciju s računalom. Odatle imamo pristup sklopkama s kojima definiramo rad upravljačke jedinice. EP-ROM memorija se može izvaditi i zamijeniti drugom koja ima drukčiji program na sebi. Također, moguće je zamijeniti bateriju koja čuva podatke pohranjene u RAM-u.



**Slika 26. Gornja strana upravljačke jedinice**

Prije nego se poveže robot s računalom, potrebno je najprije ispravno postaviti pozicije sklopki. Sklopka se smije namještati samo dok je upravljačka jedinica isključena. Njihovi nazivi prikazani su na slici 27.



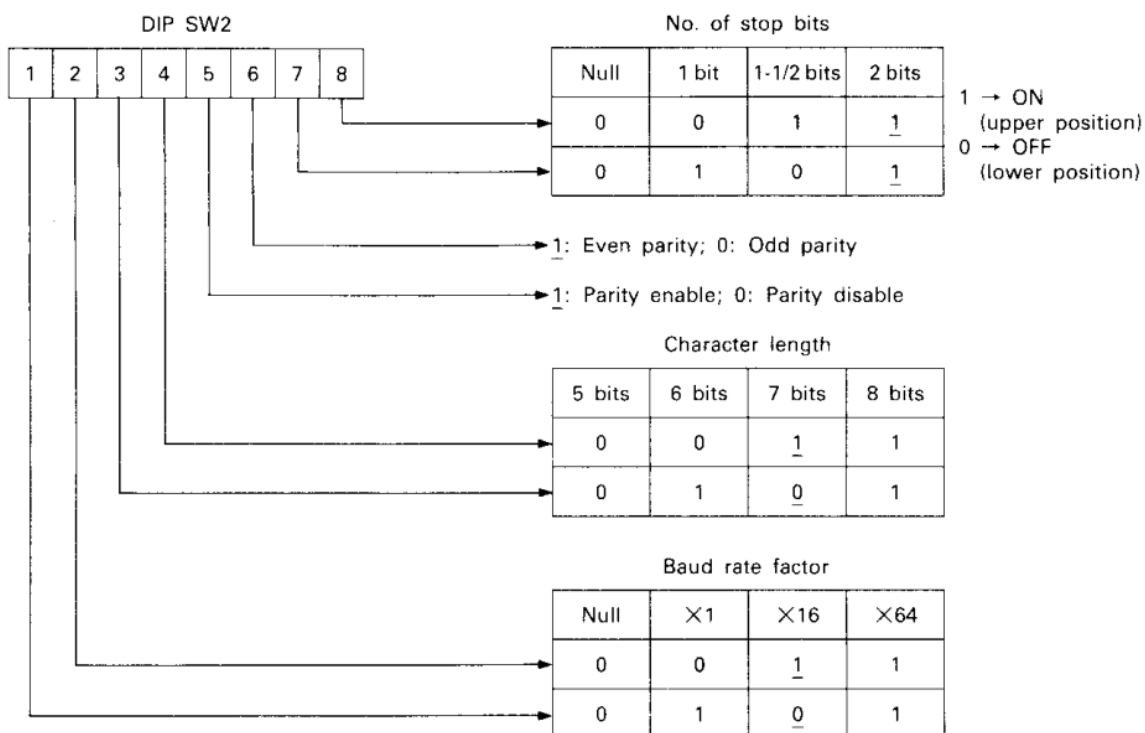
Slika 27. Sklope za definiranje rada robota

Funkcije sklopki ST1 i ST2 su sljedeće:

- ST1 – definira način rada upravljačke jedinice (gornji položaj: sustav centriran oko upravljačke jedinice; donji položaj: sustav centriran oko računala).
- ST2 – definira da li će se pri pokretanju upr. jedinice podaci iz EP-ROM-a prebaciti u RAM (donja pozicija: podaci se ne prebacuju).

Sklopke SW1, SW2 i SW3 su DIP sklopke sa po 8 pozicija/bitova. Kod sklopke SW1 navažniji za istaknut je bit 5 koji, kad postavljen u gornji položaj, omogućuje obradu signala preko I/O kartice. Time će stanica za obradu moći reći robotu kada može preuzeti obradak. U ovom položaju sklopke onemogućena su sva tipkala na prednjoj strani upravljačke jedinice, osim tipkala *Emergency stop*.

DIP sklopka SW2 i SW3 definiraju parametre serijske RS-232 veze. Slika 28. je isječak iz priručnika za tog robota koja prikazuje kako koja pozicija bita utječe na parametre veze. Podcrtane vrijednosti označavaju zadane (eng. *default*) postavke. Važno je odabrane vrijednosti koristiti kod uspostavljanja komunikacije s robotom, što će biti objašnjeno u sljedećem dijelu.



DIP SW3 Bit No.	Baud rate factor		
	×1	×16	×64
1	1200	75	—
2	2400	150	—
3	4800	300	75
4	9600	600	150
5	—	1200	300
6	—	2400	600
7	—	4800	1200
8	—	<u>9600</u>	2400

Slika 28. Postavke DIP sklopki [6]



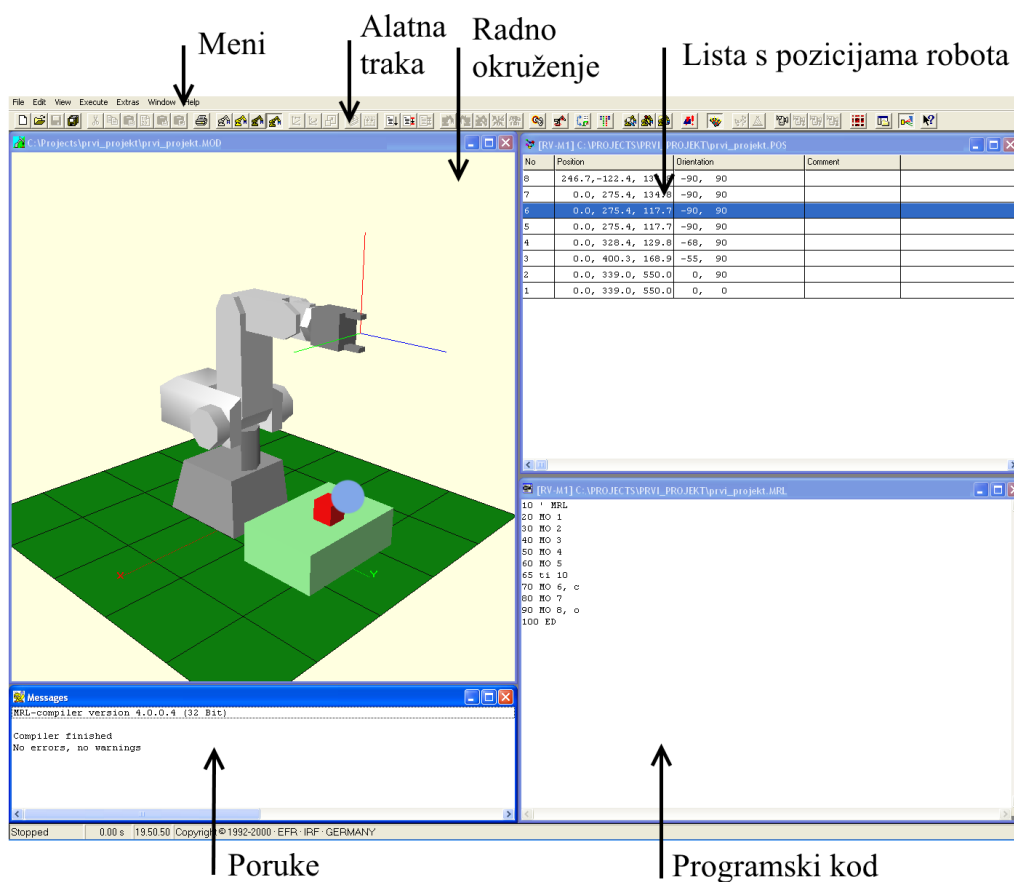
## 3.2 Programiranje robota

Programiranje robota može se, kao što je na početku spomenuto bilo, izvesti preko privjeska za učenje ili preko računala. U ovom radu je korišteno osobno računalo, a komunikacija se odvijala preko RS-232 priključka. Programski kod pisan je u MELFA BASIC jeziku koji ima jednostavnu sintaksu i moguće ga je napisati u bilo kojem uređivaču teksta.

Za komunikaciju s robotom preko računala korišten je programski alat *COSIROP*. Osim za programiranje, *COSIROP* se koristi za online kontrolu, parametrizaciju i dijagnozu Mitsubishi MELFA robota.

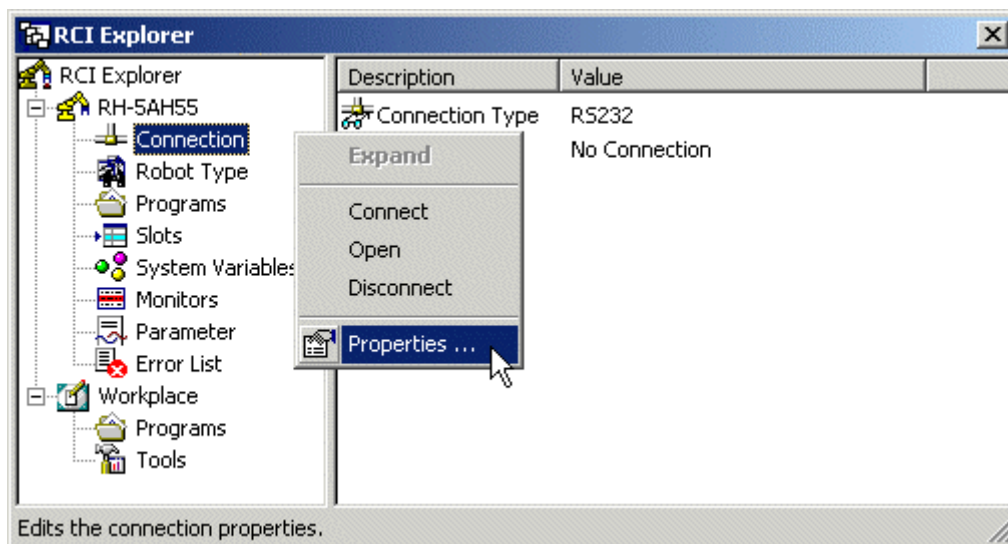
### 3.2.1 Programiranje u *COSIROP*-u

Programsko sučelje vidi se na slici 29. Radi radi lakšeg razumijevanja daljnjeg teksta, napisana su imena pojedinih dijelova sučelja. Prije nego se uključi robot potrebno je isključiti privjesak za učenje i, kao što je to prethodno objašnjeno, sklopku ST1 postaviti u donju poziciju.



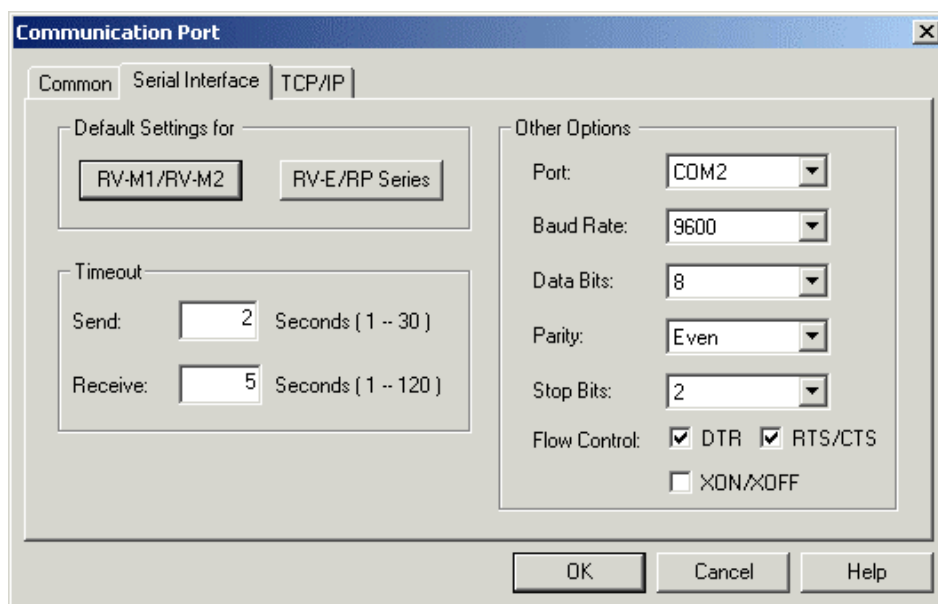
Slika 29. Korisničko sučelje *COSIROP*-a

Da bi se uspostavila komunikacija između robota i računala, potrebno je najprije postaviti dobro parametre RS-232 serijske veze. U alatnoj traci potrebno je kliknuti na *RCI Explorer* te dobijemo prozor kao na slici 30.



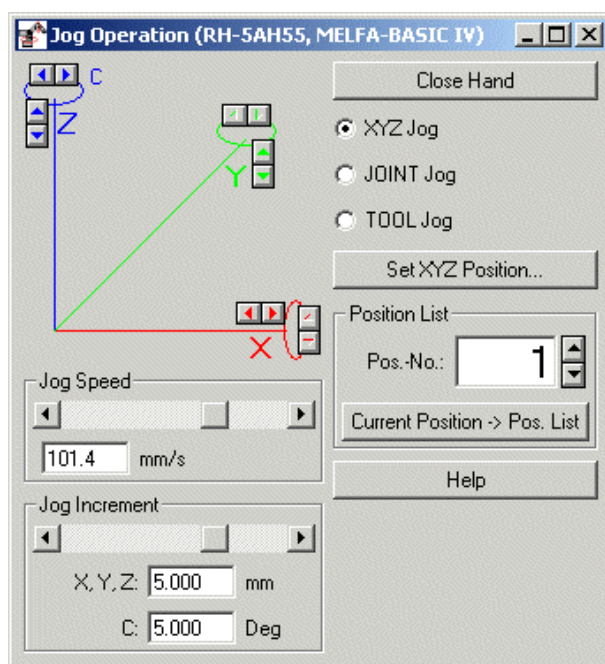
Slika 30. Postavljanje komunikacije robota [10]

Otvora nam se novi prozor u kojem treba odabrat *port* na kojem se nalazi robot, a nakon toga unijet točno definirane parametre serijske veze. Parametri veze ovise direktno o pozicijama pojedinih bitova čije je namještanje prethodno objašnjeno.



Slika 31. Parametri serijske veze [9]

Nakon što je komunikacija između robota i računala uspješno uspostavila, moguće je direktno upravljati robotom. Programski kod, kojeg ćemo poslati robotu, sastoji se od osnovnih komandi i pozicija robota. Pozicije robota se spremaju u poseban tekstualni dokumenti te se šalju robotu neovisno o programskom kodu. Da bi „naučili“ nove pozicije potrebno je najprije dovesti robota u tu poziciju te joj dodijeliti broj. Robota možemo dovesti u određenu poziciju preko privjeska za učenje ili putem računala. Na slici 32. je prikazan prozor preko kojeg možemo upravljati robotom te ga dovesti u poziciju u prostoru koja je potrebna. Također, moguće je namjestiti brzinu i veličinu inkrementa gibanja.



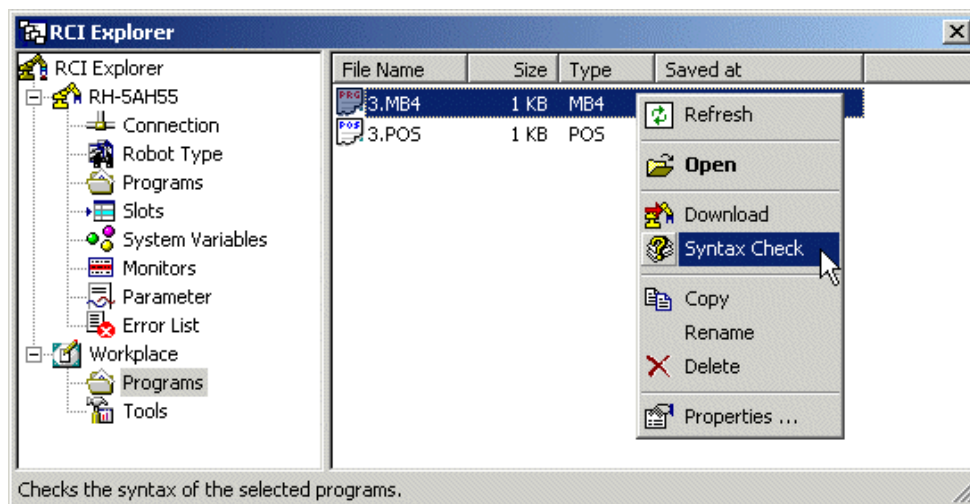
Slika 32. Prozor za upravljanje robotom [9]

Nakon što smo doveli robota u željenu poziciju potrebno je još samo kliknuti na dugme „Current Position -> Pos. List“ te će se pozicija spremi. Postupak ponavljamo za svaku poziciju. Lista s pozicijama će izgledati slično kao na slici 33.

No	Position	Orientation	Comment
P1	420.0, -260.0, 180.0	0, 0, 60,R	Position 1
P2	420.0, -260.0, 170.0	0, 0, 60,R	Position 2

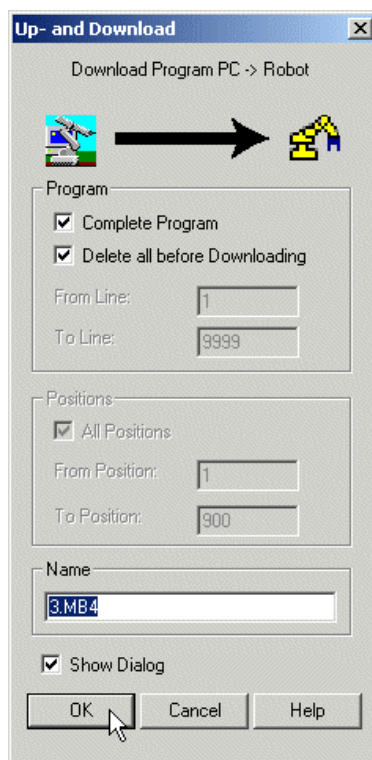
Slika 33. Lista pozicija [9]

Programski kod robota nalazi se među priložima ovog završnog rada. Još nam preostaje provjeriti sintaksu koda i izvršiti *Download* koda. Sintaksa se provjerava klikom na *Syntax check* kao što je to prikazano na slici 34.



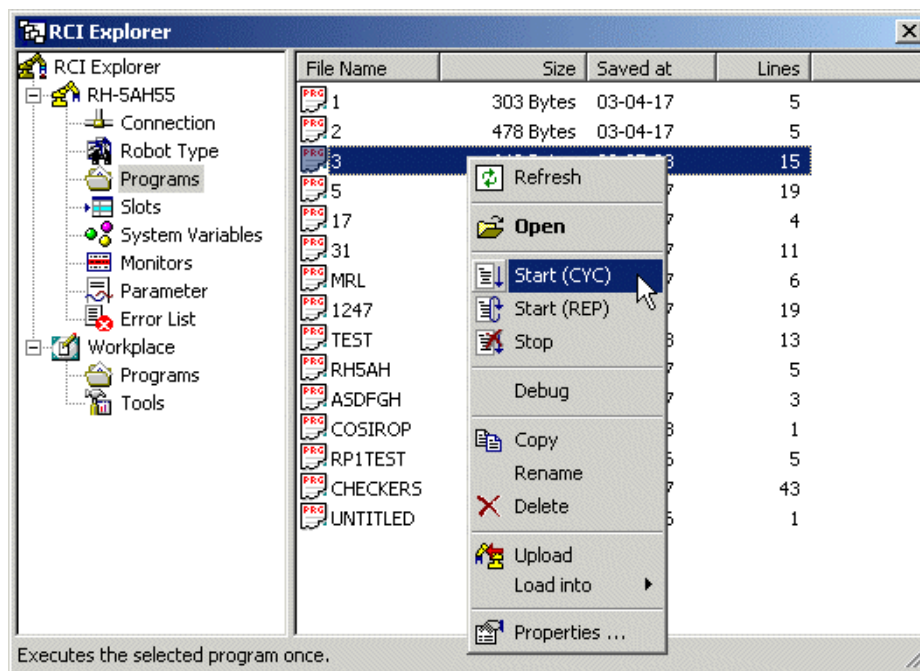
Slika 34. Provjer sintakse programa [9]

Odmah iznad tipke *Syntax Check* nalazi se tipka za *Download* koju treba potom kliknuti. Otvara se sljedeći prozor gdje je još preostalo kliknuti na *OK*.



Slika 35. Download prozor [9]

Zadnji korak jest da se program pokrene tako što se klikne na *Start (CYC)* kao što je to prikazano na slici 36.



Slika 36. Pokretanje programa [9]

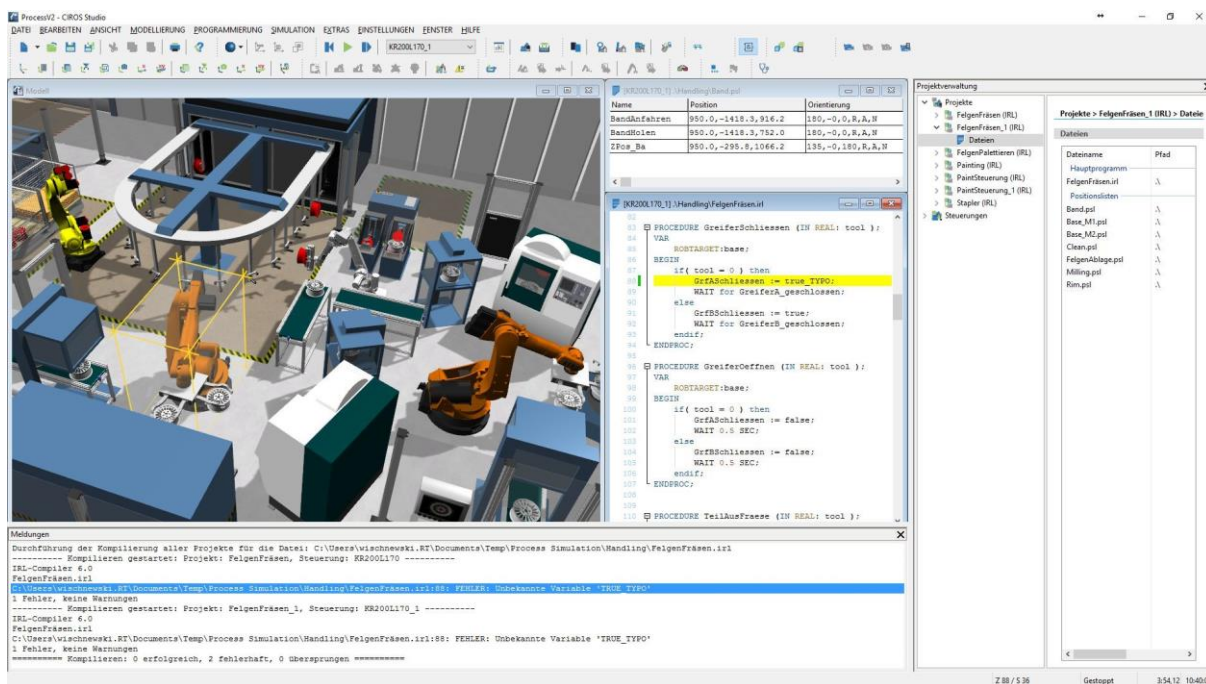
## 4. VIZUALIZACIJA U CIROS-U

Današnja računalna tehnologija omogućuje realističan 3D prikaz kompleksnih sustava automatizacije. Upotreba interaktivne vizualizacije kod sustava za automatizaciju kao što je prethodno opisana stanica, predstavlja veliki iskorak u realizaciji novih ideja. Od razvoja prototipa, planiranja postrojenja i proizvodnje, svaka faza uključuje neki vid modeliranja i vizualizacije.

U ovom završnom zadatku korišten je programski paket *Ciros Studio* 6.2.

### 4.1 Programski paket CIROS

*Ciros Studio* je profesionalni alat za vizualizaciju industrijskih okruženja kreiran i održavan od strane Festo-a. U jednom sučelju, programski paket omogućuje modeliranje i vizualizaciju stanice, a podržava i programiranje robota u različitim jezicima.



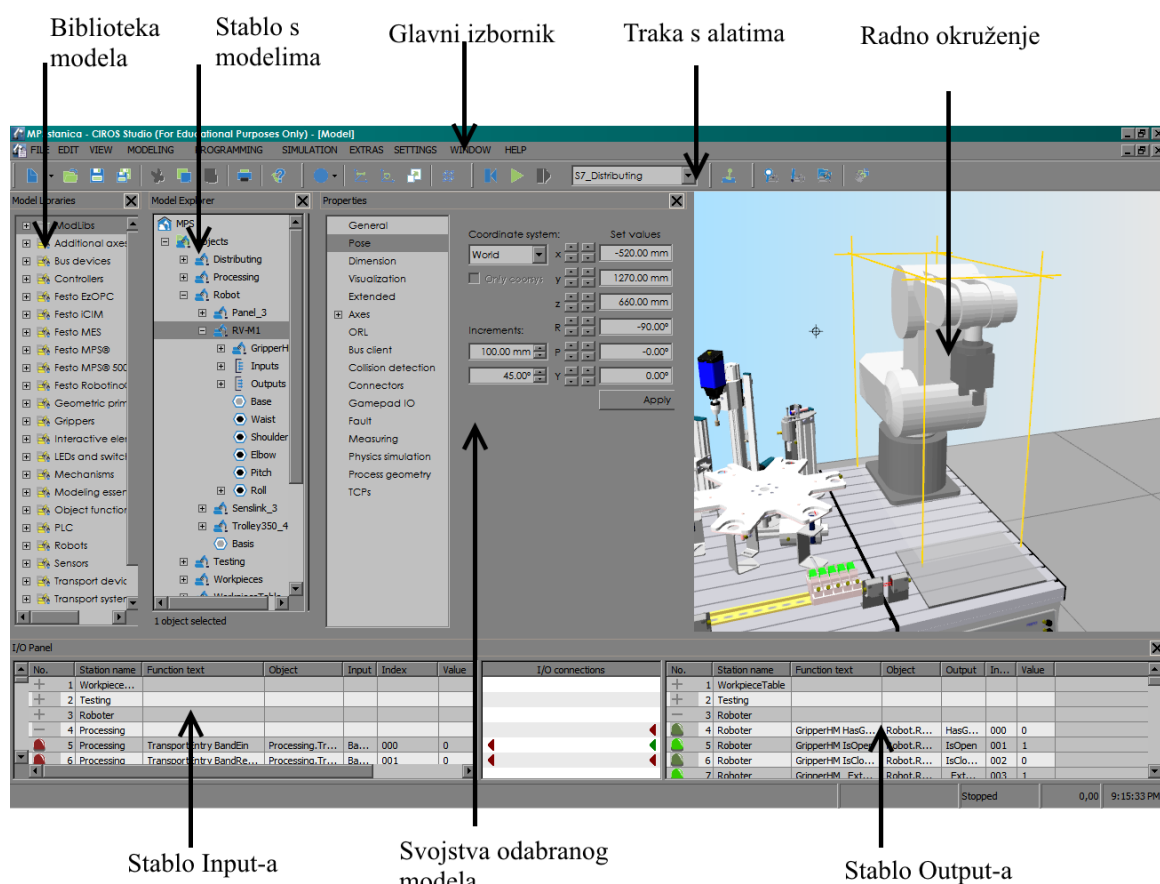
Slika 37. Korisničko sučelje u *Ciros-u* [11]

## 4.2 Kreiranje vizualizacije

Ciros dolazi instaliran s opsežnom bibliotekom gotovih modela robota, senzora, transportnih uređaja pa čak i cijelih MPS stanica. Nakon postavljanja modela u prostoru, moguće je detaljno pogledati njegova svojstva kao što su pozicija u prostoru, geometrija, odnosi s drugim elementima, I/O konfiguracija itd.

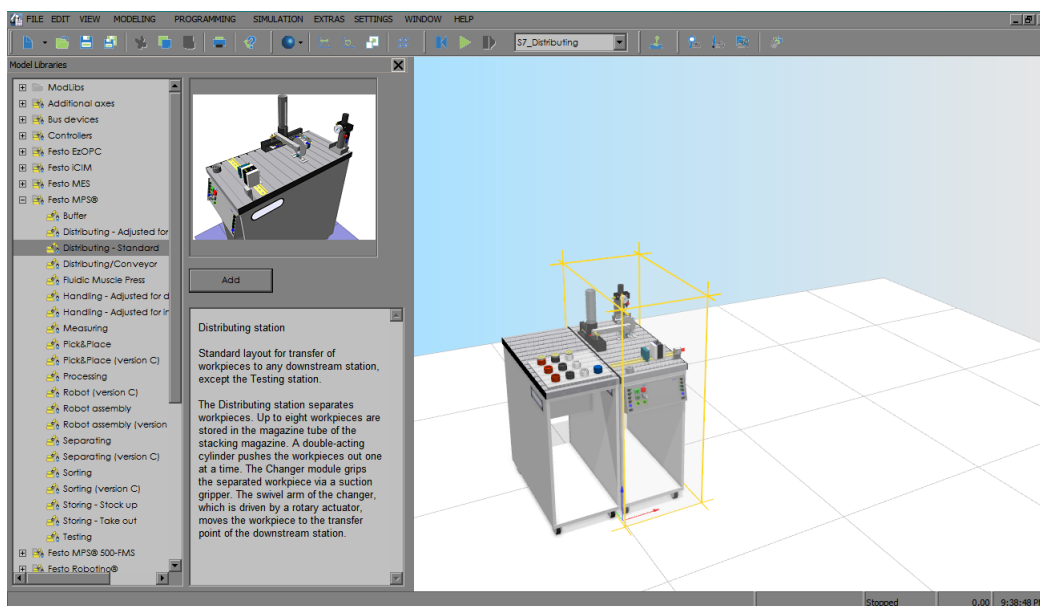
### 4.2.1 Modeliranje

Radi lakšeg razumjevanja, na slici 38. su imenovana najvažniji dijelovi sučelja koji će biti spominjani tokom izrade vizualizacije.



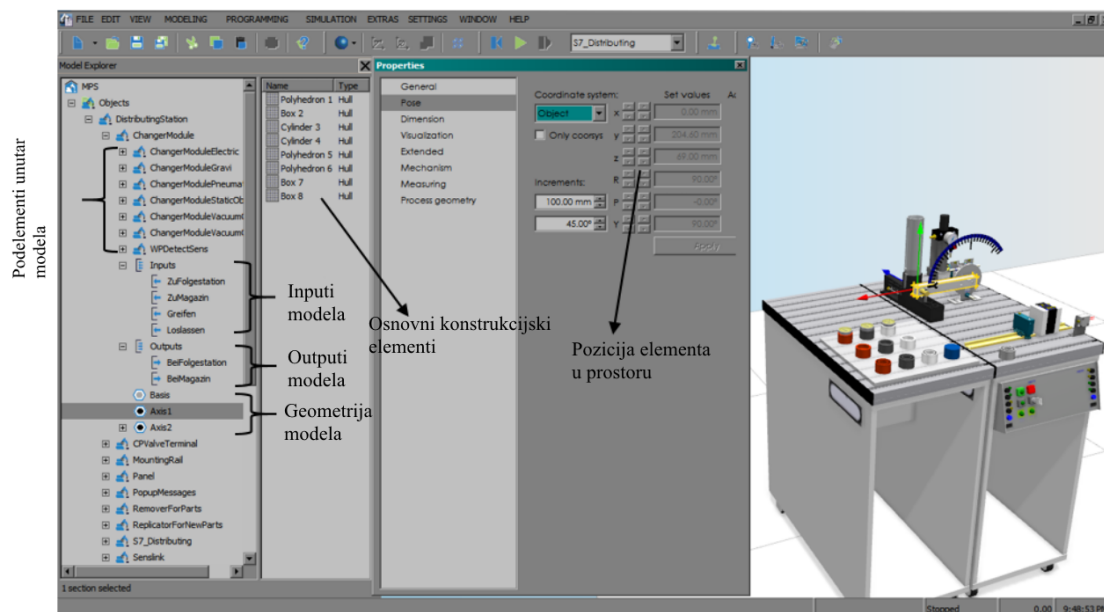
Slika 38. Dijelovi sučelja

Da bi se proizvoljni element postavio u radnom okruženju potrebno je najprije otvoriti *Biblioteku modela* pod *Modeling > Model libraries*, potom odabrati željeni model. Dobije se prozor kao na slici 39.



Slika 39. Postavljanje modela u radnom okruženju

Nova stanica automatski je postavljena uz bok prethodnoj s desne strane. Preciznije postavljanje elementa postiže se mijenjanjem Kartezijevih koordinata unutar prozora *Svojstva* (eng. *Properties*) tog elementa kao što je vidljivo na slici 40. Također, na istoj slici unutar stabla na lijevoj strani se mogu vidjeti podelementi koji su postavljeni na stanici i njihova svojstva. Svaki od njih se može pomicati u prostoru na prethodno opisani način.








Slika 40. Svojstva modela

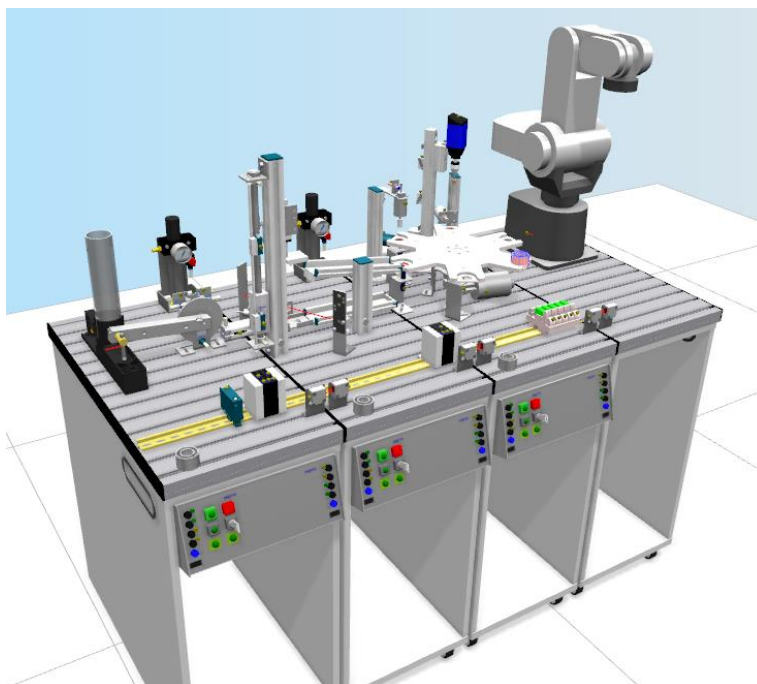


Da bi se moglo detaljno modelirati u *Ciros*-u, neophodno je imati na umu hijerarhiju tipova elemenata prikazanih u tablici 2 [10].

**Tablica 2. Hijerarhija tipova elemenata u modeliranju**

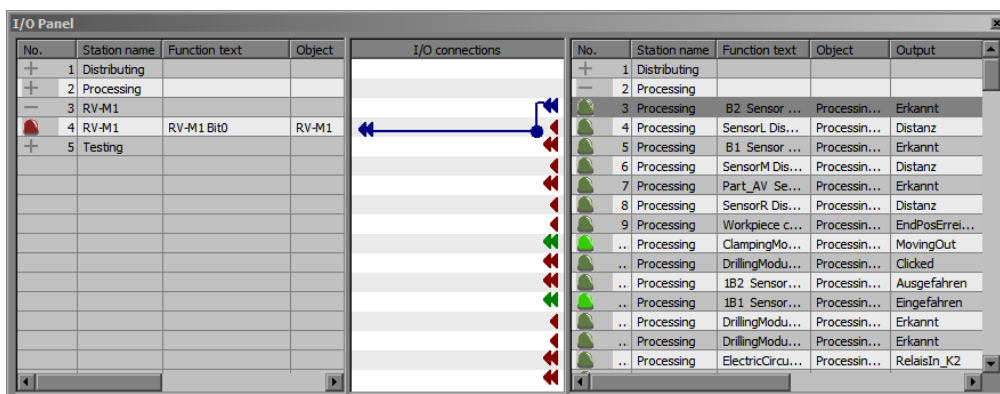
Ikona	Tip elementa	Opis	Primjer
	Objekt	Najveća jedinica u strukturi elementa	Robot
	Sekcija	Sekcije su dio objekta. Jedan stupanj slobode je asociran za svaku sekciju relativno prethodnoj sekciji	Svaki zglob robota je jedna sekcija
	Površina (eng. <i>Hull</i> )	Površine su dio sekcija i predstavljaju grafičku reprezentaciju predmeta	Sve vidljive površine
	Mjesto hvataljke (eng. <i>Gripper point</i> )	Da bi predmet bio u mogućnosti zgrabiti drugi predmet potrebno je da ima definirano mjesto hvataljke. Dodjeljuju se sekcijama.	Mjesto gdje se na robotu dolazi hvataljka
	Točka hvatišta (eng. <i>Grip point</i> )	Dodjeljuju se sekcijama. Daje mogućnost nekom predmetu da bude zgrabljen.	Svaki obradak nužno mora imati definiranu točku hvatišta

Ponavljajući prethodne korake lako se dobije cijeli MPS s robotom kao što je vidljivo na slici 41. Sve MPS stanice u *Ciros*-u dolaze s gotovim programima za PLC, a međusobnim povezivanjem one automatski uspostavljaju komunikaciju. Već u ovom trenutku je moguće pokrenuti simulaciju, no potrebno je još uspostaviti vezu između zadnje stanice i robota da bi simulacija bila potpuna.



Slika 41. Virtualna MPS stanica

Komunikacija između stanice i robota postiže se uspostavljanjem I/O veza. Otvaranjem *I/O Panel*-a pod *Menu > Modeling > I/O Panel* dobiva se sljedeći prozor kao na slici 42.



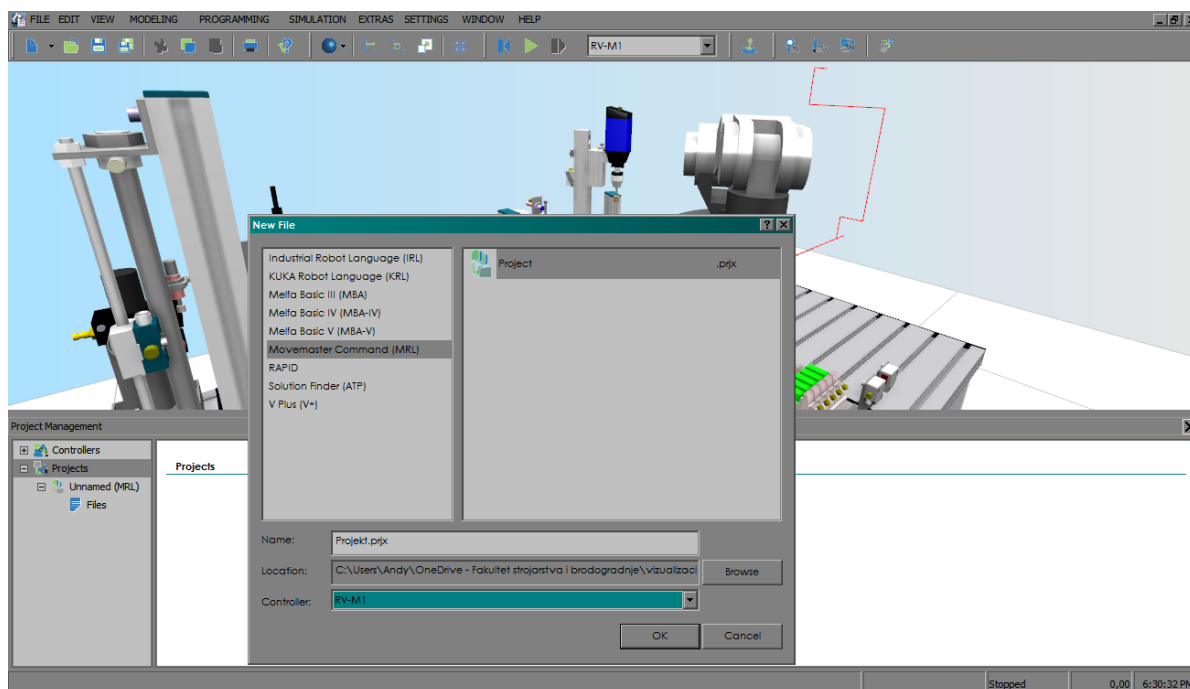
Slika 42. I/O ploča

Prozor I/O ploče podijeljen je na 3 dijela. Na lijevom dijelu nalaze se svi ulazi stanica/robota, na desnom dijelu svi izlazi, a u sredini su vizualno prikazane veze između njih. Na slici 42. je prikazana je veza između senzora i jednog ulaza na robotu. Kada se aktivira senzor koji javlja da je obradak pripremljen, robot će preuzeti i paletizirati ga. Podrazumijeva se da je prije toga potrebno napisati program za robota.

## 4.2.2 Programiranje robota

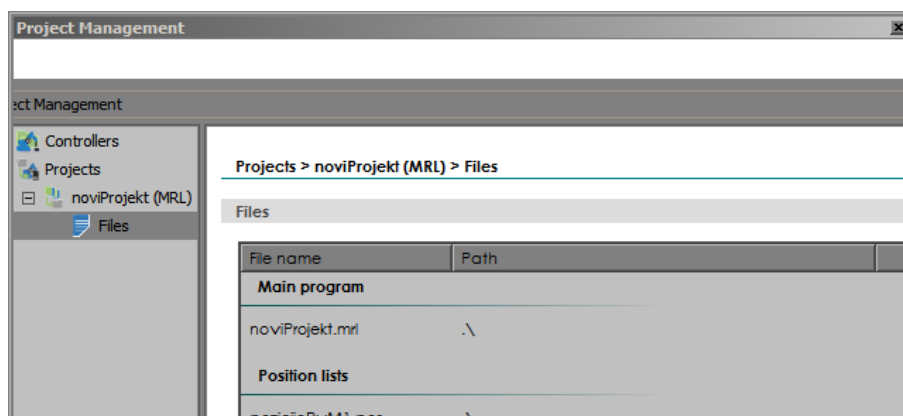
Ciros Studio nudi mogućnost programiranja Mitsubishi, KUKA, ABB, Adept i Stäubli robota te simuliranje napisanog programskog koda.

Programiranje robota započinje otvaranjem novog projekta pod *Programming > Project Management* i odabiranjem jezika koji je svojstven tom robotu. U ovom radu, pošto se koristi robot Mitsubishi RV-M1, odabran je jezik *Movemaster Command* kao što se vidi na slici 43.



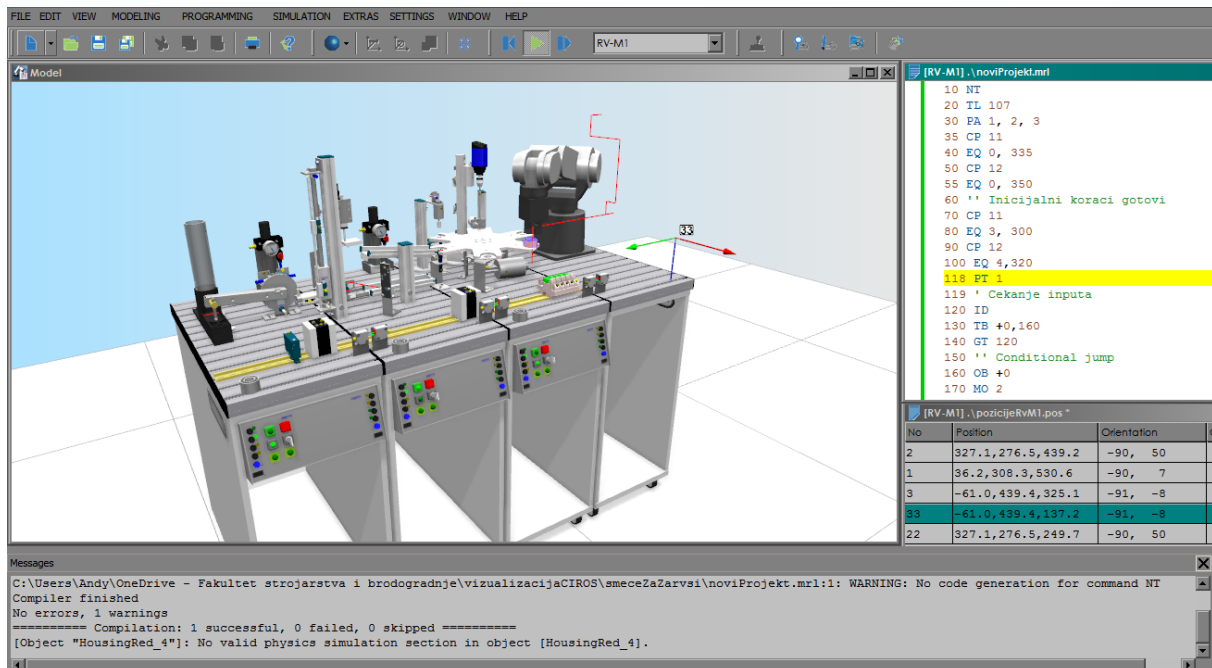
Slika 43. Otvaranje novog projekta

Novonastali projekt sastoji se od programskog koda i liste s pozicijama (slika 44.).



Slika 44. Novi projekt

Programski kod robota iz prethodnog poglavlja koristi se i ovdje. Korištenje simulacije robota u *Ciros* -u je dobar način da se provjeri funkcionalnost koda bez da se to radi na stvarnom robotu.



Slika 45. Programiranje robota

Na koncu, za pokretanje interaktivne simulacije rada stanice potrebno je još samo kliknuti na tipku *Start/Stop* u alatnoj traci.

---

## ZAKLJUČAK

Zadatak ovog završnog rada bio je zastarjele industrijske kontrolere zamijeniti suvremenim rješenjima. Pojedini dijelovi stanice zahtijevali su zamjenu oštećenih elemenata, obnovu ožičenja i druge oblike popravki. Prepoznat je kvar na robotskoj ruci, čiji popravak nije ulazio u domenu ovog završnog rada. Nadalje, dodatni cilj rada bilo je pokazati korištenje suvremenog programskog paketa za programiranje PLC-a, *Codesys*-a. Predstavljena je mogućnost izrade vizualizacije stanice u programskom paketu *Ciros*. Na koncu, važno je naglasiti kako je ovo tek prvi korak u modernizaciji stanice. Ostaje još vidjeti koje mogućnosti nudi nadolazeća industrija 4.0.

---

**LITERATURA**

- [1] Vučetić, L. – Programiranje modularnog proizvodnog sustava, Završni rad FSB Sveučilišta u Zagrebu, 2019.
- [2] Ebel F., Knoblich C. – Processing station manual, 2003.
- [3] <https://www.festo-didactic.com/int-en/learning-systems/equipment-sets/automation-technology-plc/festo-cecc-codesys-v3-compact-controller.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC4xMjMxLjc3ODA>, 19.2.2019.
- [4] Zorc, D. – Mikroprocesorsko upravljanje, Skripta s predavanja mikroprocesorskog upravljanja FSB, 2015.
- [5] <https://www.codesys.com/the-system.html>, 18.2.2019.
- [6] <http://www.roboex.com/rv-m1.PDF>, 18.2.2019.
- [7] [https://en.wikipedia.org/wiki/IEC\\_61131-3](https://en.wikipedia.org/wiki/IEC_61131-3), 16.2.2019.
- [8] [https://www.festo.com/net/et\\_ee/SupportPortal/InternetSearch.aspx?q=cecc-lk&tab=16&s=t](https://www.festo.com/net/et_ee/SupportPortal/InternetSearch.aspx?q=cecc-lk&tab=16&s=t), 19.2.2019
- [9] <http://www.provencor.fi/servo/manuals/Cosirop-firststeps.pdf>, 20.2.2019.
- [10] [https://www.festo-didactic.com/ov3/media/customers/1100/ciros\\_studio\\_manual\\_1.pdf](https://www.festo-didactic.com/ov3/media/customers/1100/ciros_studio_manual_1.pdf), 20.2.2019.
- [11] <https://www.verosim-solutions.com/en/solutions/ciros-studio/>, 20.2.2019.

## **PRILOZI**

- I. CD-R disc
- II. Tehnička dokumentacija

**PRILOG A – Programski kod RV-M1 robota**

```
10 NT
20 TL 65
30 PA 1, 2, 3
35 CP 11
40 EQ 0, 335
50 CP 12
55 EQ 0, 350
60 '' Inicijalni koraci gotovi; LOOP počinje
70 CP 11
80 EQ 3, 300
90 CP 12
100 EQ 4,320
118 PT 1
119 ' Cekanje inputa
120 ID
130 TB +0,160
140 GT 120
150 '' Conditional jump
160 OB +0
170 MO 2
180 MO 22
190 MT 22,-40, C
200 OB -0
210 ''
220 MO 1,C
230 MT 1, -50, 0
240 MO 25
250 '' Povecamo countere od palete
260 IC 11
270 IC 12
280 GT 70
290 '' Ovdje završava LOOP
300 SC 11, 1
310 GT 90
320 SC 12, 1
321 IC 3
322 CP 3
323 EQ 2,400
330 GT 118
331 '' Init
335 SC 11, 1
340 GT 50
350 SC 12, 1
360 GT 70
399 '' Paleta puna
400 OB +0
410 ID
420 TB +1,431
430 GT 410
431 SC 3, 0
432 OB -0
433 GT 70
500 ED
```