

Unaprjeđenje PLC sustava upravljanja primjenom mikrokontrolera

Kolarek, Mateo

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:083318>

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

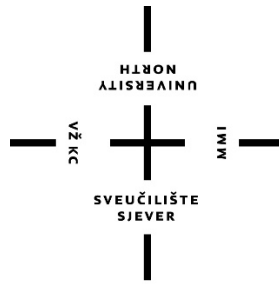
Download date / Datum preuzimanja: **2024-07-10**



Repository / Repozitorij:

[University North Digital Repository](#)





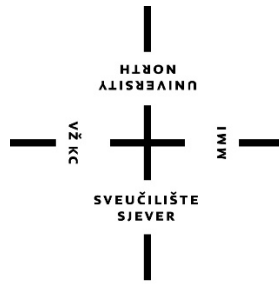
Sveučilište Sjever

Završni rad br. 478/EL/2020

Unaprjeđenje PLC sustava upravljanja primjenom mikrokontrolera

Mateo Kolarek, 2038/336

Varaždin, siječanj 2021. godine



Sveučilište Sjever

Odjel za elektrotehniku

Završni rad br. 478/EL/2020

Unaprjeđenje PLC sustava upravljanja primjenom mikrokontrolera

Student

Mateo Kolarek, 2038/336

Mentor

Josip Srpak, dipl. ing. el., viši predavač

Varaždin, siječanj 2021. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL Odjel za elektrotehniku

STUDIJ preddiplomski stručni studij Elektrotehnika

PRISTUPNIK Mateo Kolarek

MATIČNI BROJ 2038/336

DATUM 30.09.2020

KOLEGIJ PLC sustavi upravljanja

NASLOV RADA Unaprjeđenje PLC sustava upravljanja primjenom mikrokontrolera

NASLOV RADA NA ENGL. JEZIKU Improvement of PLC control system using microcontroller

MENTOR Josip Srpak dipl.ing.el.

ZVANJE Predavač

ČLANOVI POVJERENSTVA

1. mr.sc. Ivan Šumiga dipl.ing.el., viši predavač
2. doc.dr.sc. Dunja Srpak dipl.ing.el.
3. Josip Srpak dipl.ing.el., predavač
4. Miroslav Horvatić dipl.ing.el., predavač, rezervni član
- 5.

Zadatak završnog rada

BROJ 478/EL/2020

OPIS

U završnom radu potrebno je realizirati komunikaciju između razvojnog sustava baziranog na mikrokontroleru ESP32 te PLC -a FX1N-32MR. Primjenom razvojnog sustava mikrokontrolera i mobilnog telefona razviti korisničko sučelje kojim se unapređuju funkcionalnosti PLC-a. Kao primjer upravljanja izraditi idejno rješenje za projekt upravljanja samposlužnom praonicom. U realizaciji projekta koristiti WinCC SCADA sustav, te putem njega ostvariti funkcije upravljanja, nadzora i simulacije rada razmatranog projekta.

U radu je potrebno:

- opisati rješenje realizacije upravljačkog sustava,
- opisati elemente sustava,
- opisati realizaciju rješenja za komunikaciju između PLC-a, razvojnog okruženja mikrokontrolera te SCADA sustava,
- opisati i izraditi programska rješenja za svaki od pojedinih sustava,
- opisati i izraditi korisničko sučelje bazirano na ESP32 razvojnom okruženju,
- simulirati rad projekta u laboratorijskim uvjetima,
- komentirati i analizirati dobivene rezultate.

ZADATAK URUČEN

26.01.2021

POTPIS MENTORA

Josip Srpak



Predgovor

U današnje vrijeme svaka automatizacija zahtjeva što lakši i brži pristup komunikaciji sa uređajem odnosno procesom. U razmatranom slučaju PLC (eng. Programmable logic controller) uređaj komunicira s računalom na dva načina. Putem SCADA (eng. Supervisory Control and Data Acquisition) sustava vizualizacije procesa, gdje se može serijskom komunikacijom (u ovom slučaju RS232) analizirati stanje sustava u svakome trenutku, odnosno prema izrađenoj vizualizaciji promijeniti stanje rada automatizacije. Razvojnim okruženjem mikrokontrolera moguće je upravljanje putem relejnih upravljačkih jedinica, gdje se direktno može promijeniti stanje na ulazu PLC-a. To omogućuje prikaz i mogućnost upravljanja u gotovo stvarno vremensko stanje neke automatizacije uz jeftinu cijenu mikrokontrolera koji omogućuje pristup i upravljanje na daljinu preko današnjih pametnih telefona uz razvijeno korisničko sučelje.

Zahvaljujem se svojem mentoru Josipu Srpaku dipl. ing. el. kao i profesorici doc.dr.sc. Dunji Srpak dipl.ing.el. na prenosom znanju iz područja automatike i elektronike. Također se zahvaljujem u ime svih iz Sveučilišta Sjever na podršci i upornosti u radu kao i mojoj obitelji i kolegama.

Sažetak

U ovome završnom radu opisana je povijest i razvoj samog PLC (eng. Programmable logic controller) uređaja, njegov način rada te logičko upravljanje putem programskog okruženja. Kao primjer prikazuje se FX1N-32MR Mitsubishi PLC kojeg je potrebno u MELSOFT programskom okruženju logički programirati i povezati s računalom putem RS232 serijske komunikacije. Uz logičko upravljanje tzv. ljestvičastim dijagramom (eng. *Ladder Diagram, LD*) izrađuje se vizualizacija s omogućenim upravljanjem u SIMATIC WinCC razvojnom okruženju. PLC uređaj se povezuje sa vizualizacijskim alatom putem MX OPC (eng. *Mitsubishi O/I Process Control*) servera. Unutar servera se definiraju ulazi i izlazi PLC uređaja, gdje se kasnije može putem OPC (eng. *O/I Process Control*) drivera u WinCC-u učitati taj server sa ulazno-izlaznim parametrima PLC-a prethodno definiranog u MX OPC (eng. *Mitsubishi O/I Process Control*) serveru. Opisuje se analiza te kvaliteta uspostavljanja i dohvata podataka, odnosno upravljanja putem WinCC-a uz simulaciju procesa u realnom okruženju sa vizualizacijom. Govori se o prednostima te nedostacima takvog sustava i koliko je način komunikacije uistinu važan.

Opisuje se upravljanje pomoću razvojnog okruženja ESP-WROOM-32 mikrokontrolera, pri čemu se izrađuje programsko rješenje za upravljanje i dohvat podataka od strane PLC uređaja. Takvo okruženje predstavlja današnji moderni pristup, kojim je moguće upravljati gotovo u realnom vremenu. Razmatra se sigurnost protok podataka te kvaliteta i sigurnost upravljanja takvim sustavom. Izrađena je maketa idejnog rješenja u laboratorijskim uvjetima kroz rad PLC-a FX1N-32MR i paralelnim radom razvojnog okruženja ESP-WROOM-32 mikrokontrolera za upravljanje putem WiFi-a uz prikazanu vizualizaciju izrađenom u SIMATIC WinCC-u. Simulacijom rada sustava analiziraju se prednosti i nedostaci te donosi zaključak.

Ključne riječi: PLC – Programabilni logički kontroler, FX1N-32MR – Mitsubishi PLC uređaj, MELSOFT – program za programiranje programibilnih logičkih uređaja, SIMATIC WinCC – program za vizualizaciju, nadzor nekog sustava, MX OPC server, OPC driver, ESP-WROOM-32 mikrokontroler, WiFi

Summary

In this graduate thesis is description about history and development of the PLC device, his proceedings and logical management through the software systems. It is show as an example of the FX1N-32MR Mitsubishi PLC whic is necessary programmed in MELSOFT software system and connect to computer by RS232 serial communication. By the logical management so called „The ladder diagram“ (eng. Ladder Diagram, LD) developed visualization with additional, enable contron in the SIMATIC WinCC visual tools. PLC device is connected to visualization tool by the MX OPC (Mitsubishi O / I Process Control) server. Into the server, inputs and outputs of the PLC device are defined, where you can after defined load the server with the input-output parameters of the PLC previously defined in the MX OPC driver in WinCC. It is describe in the quality of establishment and retrieval of data WinCC, it can simulating process in real environment with visualization. Describes about advantages and disadvantages of such a system and how important the way of communication is.

It describes the control how using the existing ESP-WROOM-32 microcontroller, where softwre solutions are developed for the management and retrieval of data by the PLC device. Such a system represented today's modern access which operates almost in real time. There is talking about the security flow of data and about security manegement such a system. By making an example of conceptual design in laboratory working conditions of the system PLC FX1N-32MR with control and visualization by SIMATIC WinCC and Parallel operation existing ESP-WROOM-32 controller whic is controled by WiFi. The system simulation shows the advantages and disadvantages of the same and analyzes the management process.

Keywords: PLC – Programmable logic controller, FX1N-32MR – Mitsubishi PLC device, MELSOFT – software for programming Mitusbishi PLC devices, SIMATIC WinCC – software for visualization, supervison of data (SCADA tool), MX OPC server, OPC driver, ESP-WROOM-32 microcontroller, WiFi

Popis korištenih kratica

ACK	Acknowledgment – Slanje potvrde u namjenskoj vezi
BMP180	Barometric Pressure sensor – Jedan od BMP serija senzora za mjerenje tlaka i temperature
COM	Communication port – Komunikacijski port
CNC	Computerized Numerical Control – Računalno numerička kontrola
CPU	Central Processing Unit – Središnja procesorska jedinica
CSS	Cascading Style Sheets – Stilski jezik za uređivanje web dokumenata
ENQ	Enquiry – Slanje upita u namjenskoj vezi
ERP	Enterprise Resource Planning – Planiranje resursa
ESP32	Seriya mikrokontrolera strane tvrtke Espressif Systems
ESP8266	„Nasljednik ESP32 mikrokontrolera“
ETX	End-of-Text-Character – Kodni karakter kraja zapisa
FBD	Function Block Diagram – Funkcijski blok dijagram
FX1N	Seriya PLC uređaja tvrtke Mitsubishi
GPIO	General Purpose Input/Output – Ulaz/izlaz opće namjene
HMI	Human-Machine Interface – „Ljudsko sučelje stroja“
HTML	Hypertext Markup Language – Prezentacijski jezik za izradu web stranica
I/O	Input/Output – Ulaz/Izlaz
IoT	Internet of Things – „Internet stvari“
IP	Internet Protocol – Internet protokol
LD	Ladder diagram – Ljestvičasti dijagram
MES	Manufacturing Execution System – Sustav izvršenja proizvodnje
MELSOFT	Mitsubishi Electric brand Software - Mitsubishi PLC program
MX OPC	Mitsubishi O / I Process Control – Mitsubishi ulazno-izlazna kontrola procesa
NO	Normally open contact – Normalno otvoreni kontakt
NC	Normally closed contact – Normalno zatvoreni kontakt
OLED	Organic light-emitting diode – „Organska“ svjetleća dioda
PLC	Programmable logic controller – Programabilni logički kontroler
RAM	Random Access Memory – RAM memorija
ROM	Read only Memory – ROM memorija
RS-232	Recommended Standard of Communication 232 – Standard komunikacije 232
RS-485	Recommended Standard of Communication 485 – Standard komunikacije 485

SCADA	Supervisory Control and Data Acquisition – Sustav za vizualizaciju te nadzor upravljanja nekog sustava
SSR	Solid State Relay – Poluvodički relej
STEP7	Program za programiranje Siemens PLC uređaja
STL	Statement List – Instrukcijski list PLC programa
STX	Start-of-Text-Character – Kodni karakter početka zapisa
WINCC	SCADA and HMI from Siemens – SCADA sustav tvrtke Siemens
WIFI	Wireless Fidelity – Bežična mreža

Sadržaj

Uvod.....	8
1. PLC – programabilni logički kontroler.....	10
1.1. Općenito	10
1.2. Povijest i razvoj uređaja	10
1.3. Princip rada PLC-a	12
1.4. Struktura PLC-a.....	13
1.5. CPU modul i memorija PLC-a	13
1.6. Vrste PLC-a.....	14
1.7. Programiranje PLC uređaja.....	15
1.8. PLC FX1N – 32MR	17
1.9. Programsko sučelje MELSOFT series GX Developer.....	18
1.10. Povezivanje PLC-a sa računalom.....	19
1.11. Izrada PLC programa	20
1.12. Simulacija PLC programa	26
1.13. Prednosti i nedostaci PLC uređaja	28
2. SCADA – Supervisory Control and Data Acquisition	29
2.1. Općenito	29
2.2. Povijest i razvoj SCADA sustava.....	29
2.3. Komunikacija PLC-a i SCADA sustava	30
2.4. WinCC Explorer.....	34
2.5. MX OPC Configurator – server	35
2.6. Povezivanje WinCC-a sa PLC uređajem FX1N-32MR.....	39
2.7. Graphics Designer	40
2.8. WinCC Runtime simulacija	49
2.9. Karakteristike SCADA sustava.....	53
3. Upravljanje sustava pomoću ESP-WROOM-32.....	54
3.1. ESP-WROOM-32.....	54
3.2. Upravljanje pomoću ESP-WROOM-32.....	55
3.3. Povezivanje ESP-WROOM-32 sa FX1N-32MR	55
3.4. Programsko rješenje	57
3.5. Komunikacija ESP32 i klijentsko-poslužiteljska veza.....	64
3.6. Izgled pristupnog web sučelja	66
3.7. Unaprjeđenje rada ESP32 pomoću BMP180	68
3.8. Karakteristike ESP32 mikrokontrolera	71
4. Shema sustava upravljanja	73
5. Simulacija sustava.....	75
5.1. Izgled sustava upravljanja	75
6. Zaključak.....	77
7. Literatura.....	78

Uvod

Ideja završnog rada bila je implementirati sustav koji je jednostavan za upravljanje na više načina, npr. sustav kojeg je moguće upravljati putem mikrokontrolera s izrađenim razvojnim okruženjem na nekom pametnom telefonu ili izradom vizualizacije procesa s mogućnošću upravljanja u SCADA (eng. *Supervisory Control and Data Acquisition*) sustavu. Takav sustav zahtjeva detaljnu obradu svih parametara te svakog uzroka koji može stvoriti problem, a to se rješava alarmnim metodama (obavijestima) ili uvjetima isključenja.

Računala su službeno predstavljena 1980-tih godina, kada je industrijska automatizacija dostigla vrh kod proizvodnje automobila. Industrijska automatizacija je zapravo automatizirano upravljanje poput korištenja PLC-a (eng. *Programmable logic controller*) ili računalno-numeričkog upravljanja (CNC) . To je zapravo pripomoć kontroli i procesiranju u industrijama, što zamjenjuje ručne intervencije koje čak mogu dovesti čovjeka u opasnost.

U svijetu se gotovo sve više koristi industrijska automatizacija od niza elemenata koji su sinkronizirani te koji izvršavaju funkcije poput upravljanja, sensoriranja, nadzora te praćenja procesa. Funkcionalno gledano, industrijska automatizacija uključuje razinu upravljanja nad radom procesa pomoću senzora ili aktuatora, nadzor te razinu kontrole pomoću SCADA sustava i slanjem informacija procesa (MES i ERP) .

U ovome završnome radu detaljno se analizira takav automatizirani sustav putem vizualizacije SIMATIC WinCC-a i upotrebom razvojnog okruženja ESP-WROOM-32 mikrokontrolera kao „prijenosnikom“ podataka za nadzor, odnosno upravljanje sustava putem serijske komunikacije kod FX1N-32MR PLC-a (eng. *Programmable logic controller*) .

Današnje automatizacije se sve češće izrađuju na način upravljanja putem različitih IoT (eng. *Internet of Things*) razvojnih sustava, koje imaju mogućnost upravljanja putem WiFi-a ili Bluetooth-a. Takvo korištenje omogućuje brže djelovanje na sustav te jednostavnu analizu prilikom upravljanja i prikupljanje podataka za nadzor. PLC je u današnje vrijeme veoma bitan faktor kod upravljanja sustava i vrlo je važno imati brzi protok podataka kod nadzora ili upravljanja. Time je izrađen primjer realizacije automatiziranog sustava, kojim se može upravljati na dva različita načina. Prilikom simulacije izrađene makete analizirat će se kvaliteta rada, upravljanje te prednosti i nedostaci takvih sustava.

Automatizirani sustav upravljanja temeljen je na FX1N-32MR PLC-u sa dvije upravljačke cjeline, a to su SCADA (eng. *Supervisory Control and Data Acquisition*) sustavom te sustavom razvojnog okruženja ESP-WROOM-32 mikrokontrolera pomoću pametnog telefona kojim je moguće upravljati. Smisao takvog sustava jest povećanje sigurnosti te samog nadzora nad istim, da radi što pristupačnije i točnije.

Kada se govori o sigurnosti važno je kod ozbiljnih automatiziranih sustava da su osigurani od mogućih oštećenja prilikom kvara te osiguranje od mogućih softverskih kvarova (djelovanje zlonamjernog softvera ili loše programiranog sustava) koji mogu prouzročiti nepravilan rad, ali i dovesti do kvara nad sustavom. U slučaju nastanka kvarova bitna je pristupačnost da se kvar što brže pronađe te otkloni uz naznaku imati rezervne komponente koje su „osjetljivije“ na nestabilnost i češće su sklone kvaru. Zbog toga sustavi kompleksnije automatike moraju imati češće provjere te detaljan nadzor.

1. PLC – programabilni logički kontroler

1.1. Općenito

PLC (eng. *Programmable logic controller*) je kratica za „Programabilni logički kontroler“ što je zapravo računalo dizajnirano za rad u industrijskim postrojenjima. Koriste se za automatizaciju industrijskih procesa npr. kod montažnih linija nekog pogona, uređaja kod pročišćavanja otpadnih voda ili u nekim drugim sustavima. PLC dijeli mnoge značajke kao i računalo. Oboje imaju napajanje, CPU (eng. *Central processing unit*), ulaze i izlaze (I / O), memoriju te operacijski softver. Najveća razlika je u tome što PLC može izvršavati diskretne i kontinuirane funkcije koje računalo ne može raditi, a time je PLC puno prikladniji za „gruba“ industrijska okruženja, smatra se kao „robustnim“ digitalnim računalom koji upravlja elektromehaničkim procesima u nekom industrijskom okruženju.



Slika 1.1 Prikaz PLC uređaja tip Siemens Simatic S7-200 [1]

1.2. Povijest i razvoj uređaja

Prije „rođenja“ PLC uređaja jedini način upravljanja strojevima bio je korištenjem releja. Releji rade pomoću zavojnice, kojoj kad se inducira napon (Elektromotorna sila) stvara magnetsko polje koje svojom silom povlači kotvu odnosno kontakt kao prekidač u položaj uklopa ili isklopa. Isklop se dogodi u slučaju kada se relej isključi i time se kontakt otpušta u prvobitan položaj te je time kontakt otpušten i ne teče struja. Moguća je izrada relejnih kontakata normalno otvorenih (NO) ili zatvorenih (NC).

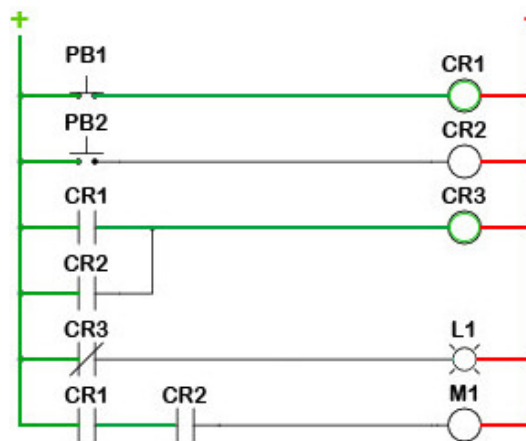
Tim načinom korištenja releja u industrijama se stvaralo puno mrežnih releja koji su upravljali industrijskim procesima (Radom strojeva ili slično). Takvo „gomilanje“ elektromehaničkog upravljanja nije bilo ekonomično u to vrijeme. Jedan kvar releja mogao bih prouzročiti prestanak rada sustava, a otklanjanje kvara bi trajalo veoma dugo zbog ogromne „mreže“ ostalih releja koji su povezani.

Izum PLC uređaja dogodio se 1960/70-tih godina strane *Dicka Morleya* koji je doveo revoluciju u industrijskoj proizvodnji.

Mogućnosti prvog PLC-a :

- Fleksibilnost upravljačkih releja (Solid-state) poput računala sa pristupačnim cijenama kao i cijenom elektromehaničkih releja.
- Jednostavno održavanje te programiranje u skladu već prihvaćenom logikom
- Omogućen rad u industrijskom okruženju (nečistoća, vlaga, vibracije ..)
- Mogućnost modularnog oblika zbog jednostavne izmjene komponenata te proširivanjem

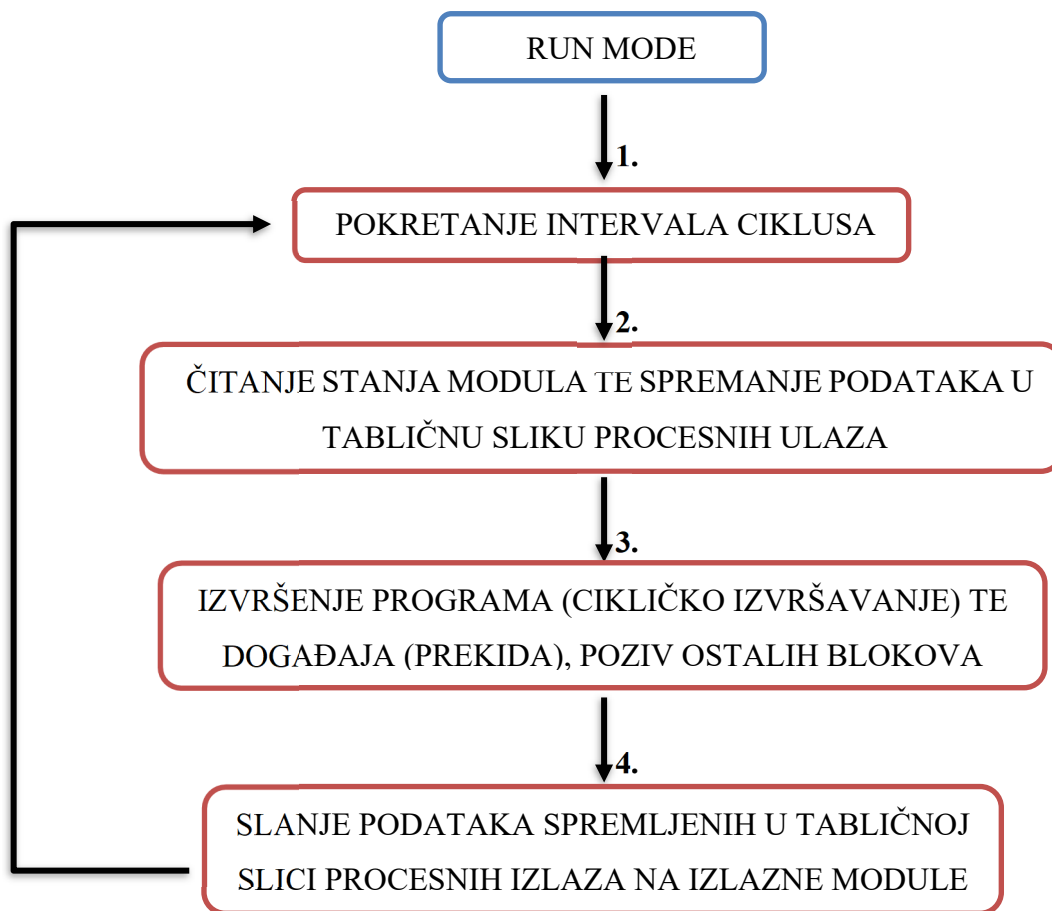
Programski izgled PLC-a zahtijevao je da ga električari za održavanje i inženjeri postrojenja lako razumiju i koriste. Kako su se upravljački sustavi temeljeni na relejima razvijali i postajali sve složeniji, upotreba shema ožičenja mjesta fizičkih komponenata također je evoluirala u logiku releja koja se pokazuje ljestvičasto, a time je donesena odluka o vrsti programiranja PLC uređaja.



Slika 1.2 Ljestvičasti prikaz programiranja PLC-a [3]

1.3. Princip rada PLC-a

Rad PLC-a najjednostavnije se može prikazati cikličkom metodom.

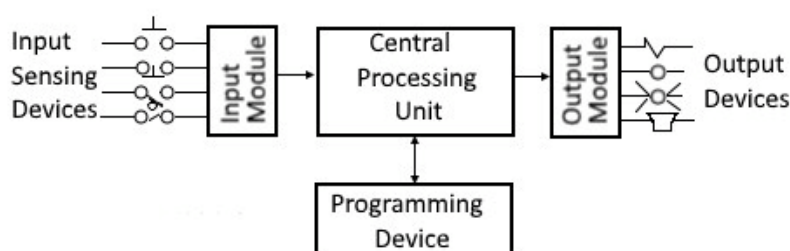


1. Operativni sustav započinje ciklus te nadgleda vrijeme.
2. CPU započinje čitanje podataka iz ulaznog modula i provjerava status svih ulaza.
3. CPU započinje s izvršavanjem korisničkog ili aplikacijskog programa napisanog u relejnoj ljestvici ili bilo kojem drugom PLC programu.
4. CPU dalje izvodi sve interne dijagnostičke i komunikacijske zadatke i prema rezultatima zapisuje podatke u izlazni modul time se svi izlazi ažuriraju.

1.4. Struktura PLC-a

Programabilni logički kontroleri kontinuirano nadziru ulazne vrijednosti iz različitih ulaznih senzorskih uređaja te proizvode odgovarajuće rezultate ovisno prema PLC programu. Tipični PLC uređaj sastoji se od 5 glavnih dijelova:

1. Kućište
2. Modul napajanja
3. CPU
4. Ulazno-izlazni modul
5. Modul komunikacijskog sučelja



Slika 1.3 Prikaz blok dijagrama PLC uređaja [4]

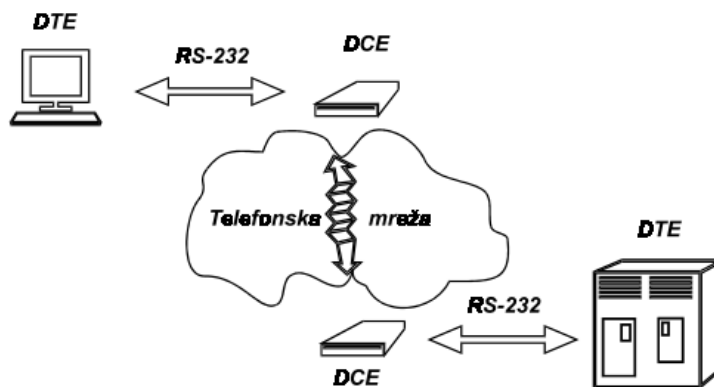
1.5. CPU modul i memorija PLC-a

CPU (eng. *Central processing unit*) ima središnji procesor, ROM i RAM memoriju. ROM memorija uključuje operativni sustav, upravljačke programe i aplikacijske programe, a RAM memorija se koristi za pohranu programa i podataka. CPU je „mozak“ PLC-a sa oktalnim ili heksadekadskim mikroprocesorom.

Kod PLC-a postoji modul za povezivanje ulaza i izlaza te se naziva ulazno-izlazni modul. Ulazi mogu biti tipkala za pokretanje ili zaustavljanje, prekidači itd., a izlazne komponente mogu biti ventili, releji, žarulje i sl. , Ulazno-izlazni modul pomaže kod povezivanja ulaznih i izlaznih komponenti s mikroprocesorom.

Za prijenos podataka između CPU-a i računala postoje serijski priključci koji se mogu vrlo jednostavno povezati pomoću kabela, odnosno vrstom komunikacijskog protokola kao što su RS-232 te RS-485.

RS-232 je zapravo standard koji definira digitalnu tj. serijsku komunikaciju između DTE (eng. *Data Terminal Equipment*) i DCE (eng. *Data Communication Equipment*). Izvorno je zamišljen za povezivanje računala i terminala (DTE) s modemima (DCE) za potrebe udaljenog rada na središnjim računalima. RS-232 je ograničen s obzirom na brzinu prijenosa i udaljenosti. Maksimalna brzina prijenosa je 20kbps na udaljenosti od 15m . Također drugi nedostatak takve komunikacije je način komuniciranja kao point-to-point izvedba koja omogućuje komunikaciju samo s jednim uređajem.



Slika 1.4 Prikaz povezivanja digitalne komunikacije standarda RS-232 [5]

RS-485 je standard digitalne komunikacije koji se masovno primjenjuje u industriji kada je potrebno ostvariti put za prijenos digitalnih podataka između više sugovornika. RS-485 je pouzdan te se može postići brzina od 100kbps za 1200m, tj. 10Mbps za 20m pa čak i veće do 25Mbps. RS-485 koristi za prijenos podataka dvije linije (A i B) kroz koje se istovremeno prenosi isti podatak, a informacija na strani prijemnika da li je podatak binarna jedinica ili ne, dobije se razlikom napona između linija A i B.

1.6. Vrste PLC-a

Postoje dvije glavne vrste PLC-a :

- Fiksni/Kompaktni PLC – ima fiksni broj ulaza i izlaza, nema mogućnost proširenja modula
- Modularni PLC – omogućuje višestruko proširenje kroz „module“, ulazno izlazne komponente se mogu povećati, jednostavniji je za upotrebu jer je svaka komponente neovisna jedna o drugoj.

Također se PLC može podijeliti prema vrsti izlaza:

- Relejni izlaz – najprikladniji za AC/DC izlazne uređaje
- Tranzistorski izlaz – koristi sklopne operacije te se koristi unutar mikroprocesora

Prema fizičkoj veličini PLC može se podijeliti na :

- Mini PLC
- Mikro PLC
- Nano PLC

Neki od proizvođača PLC-a :

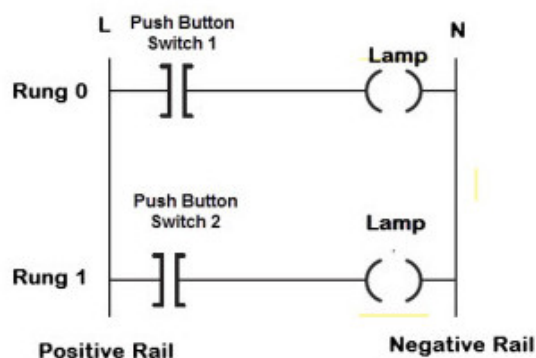
- Allen Bradley
- ABB
- Siemens
- Mitsubishi PLC
- Hitachi PLC
- Delta PLC
- General Electric (GE) PLC
- Honeywell PLC

1.7. Programiranje PLC uređaja

Kod korištenja PLC uređaja važno je osmisliti i implementirati koncept rada PLC-a prema namjeni. Da bismo to učinili, mora se razumjeti logika PLC programiranja.

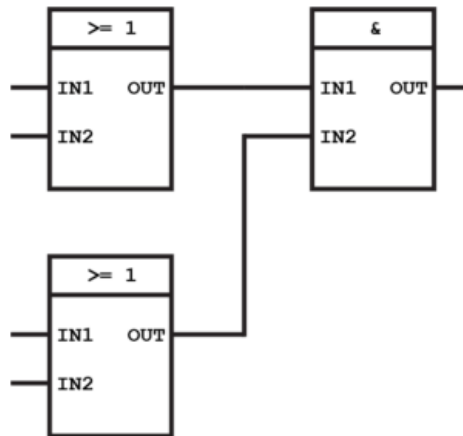
Postoje tri najčešće logike programiranja PLC uređaja:

- Ladder dijagram – najjednostavniji oblik PLC programiranja, poznati kao „relejna logika“



Slika 1.5 Prikaz PLC programiranja Ladder dijagramom [4]

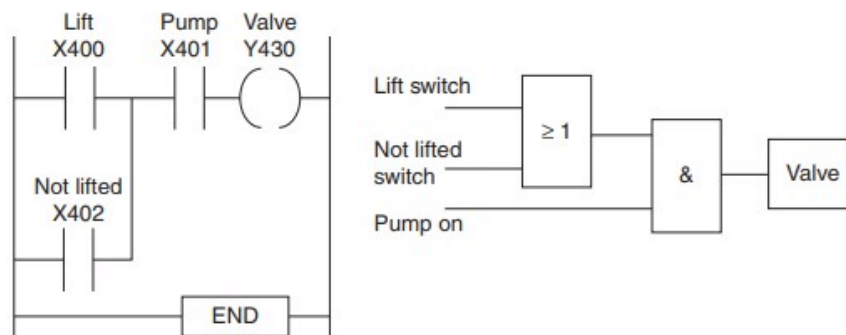
- FBD – funkcijski blok dijagram – jednostavna grafička metoda za programiranje više funkcija u PLC-u



Slika 1.6 Prikaz PLC programiranja FBD-om [4]

- STL – eng. STATEMENT LIST – tekstualni programski jezik koji pomoću izraza određuje što se treba izvršiti. Niz se sastoji od opisa zadatka te operatora

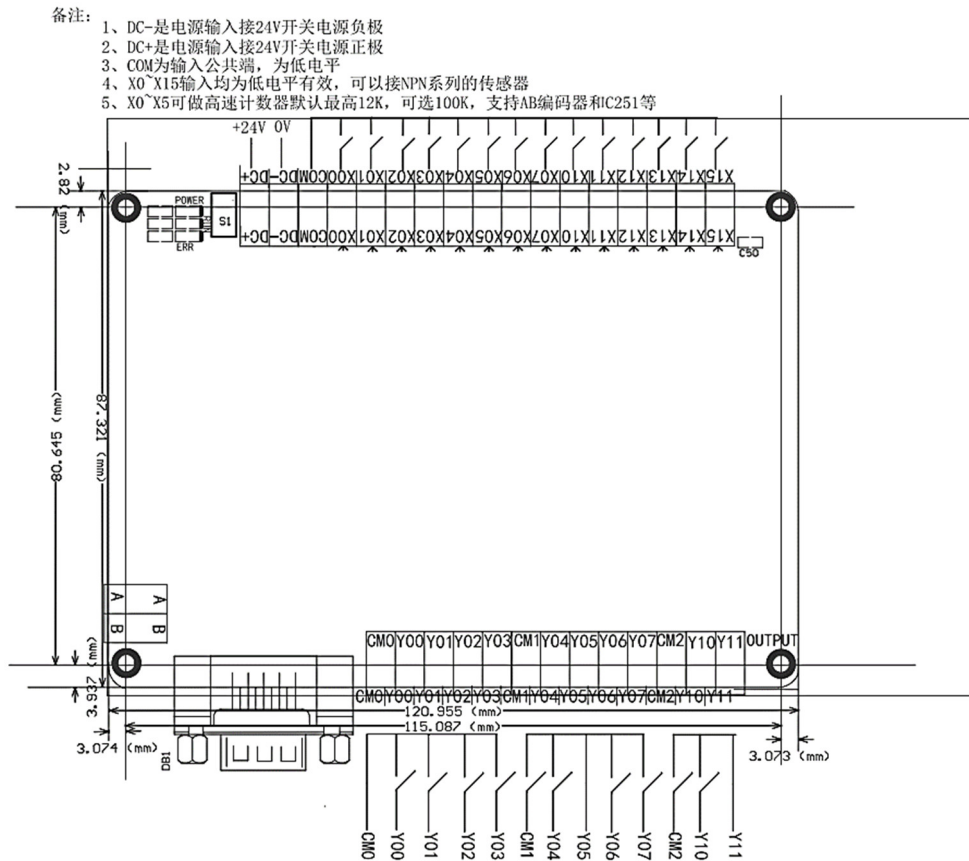
U nastavku je prikazan identičan program programiran u Ladder-u te FBD-u.



Slika 1.7 Primjer programiranja Ladder-om (lijevo) te FBD-om(desno) [4]

1.8. PLC FX1N – 32MR

U radu kao primjer izrade sustava uzet je PLC uređaj FX1N-32MR Mitsubishi sa 16 ulaza i 16 izlaza. Ima mogućnost povezivanja preko RS-232 digitalne komunikacije. Fiksni je, nema mogućnost modularne ugradnje te je izveden relejnim izlazom.



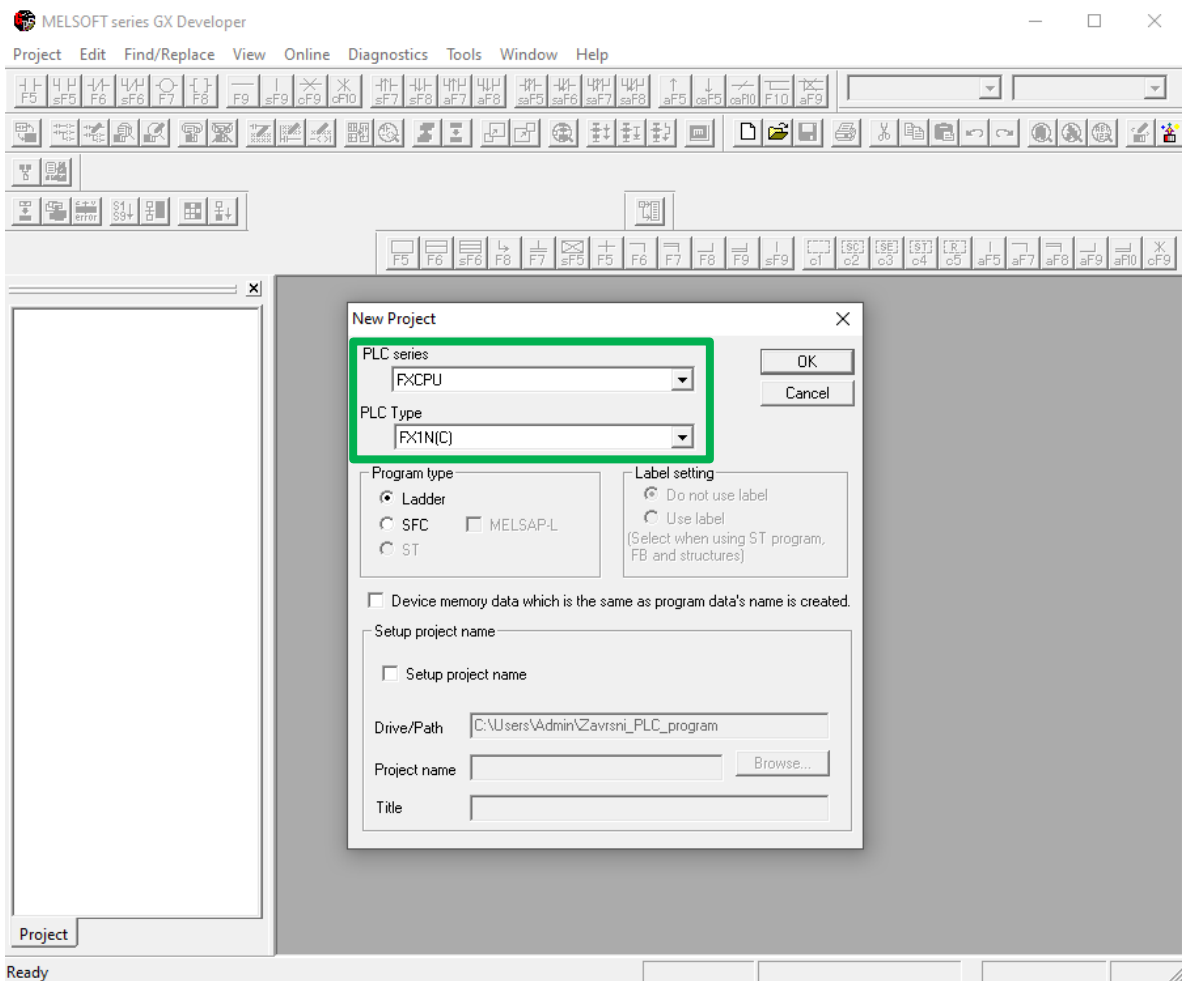
Slika 1.8 Prikaz blok sheme FX1N-32MR PLC-a prema proizvođaču



Slika 1.9 Prikaz izgleda FX1N-32MR PLC-a [7]

1.9. Programsko sučelje MELSOFT series GX Developer

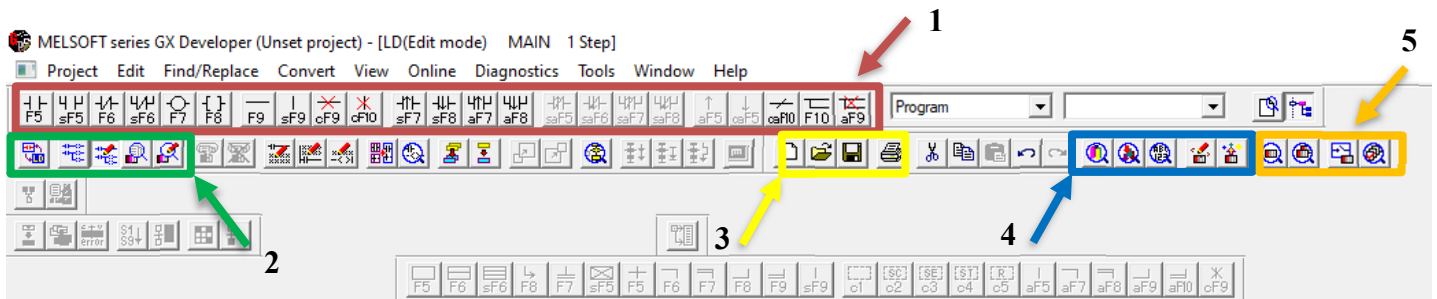
FX1N-32MR moguće je programirati MELSOFT programskim paketom prema proizvođaču Mitsubishi-u.



Slika 1.10 Prikaz izgleda MELSOFT programskog sučelja sa izradom novog projekta

Otvaranjem programa odabere se novi projekt te odredi se serija i tip PLC uređaja i pri tome način programiranja.

Prikazat će se programski sustav praonice s automatskim i ručnim režimom rada koji će odgovarati kasnijoj vizualizaciji procesa i unaprjeđenju mikrokontrolera.



Slika 1.11 Prikaz radne trake sa osnovnim funkcijama MELSOFT programa

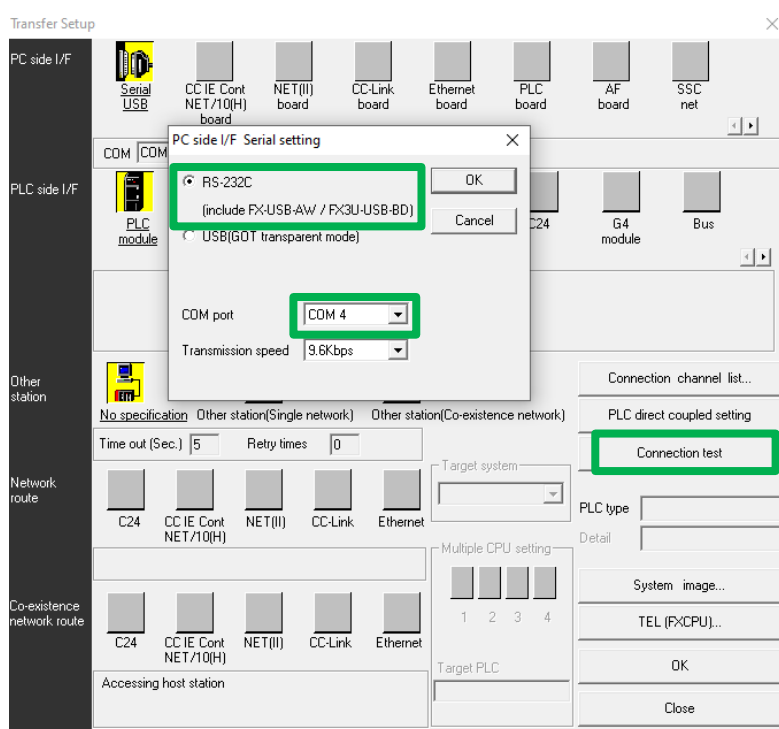
Opis označenih dijelova radne trake MELSOFT programa

1	Prikaz osnovnih simbola te oznaka za izradu Ladder programa (Otvoreni kontakt, zatvoreni kontakt, oznaka izlaza zavojnice releja, ulazi sa pozitivnim ili negativnim bridom, dodavanje linija za povezivanje, brisanje linija)
2	Prikaz različitih pogleda programa prema modelu (Pregledni, radni, monitoring, monitoring s mogućim mijenjanjem programa)
3	Mogućnost otvaranja starog ili izrade novog projekta, spremanje projekta, ispis projekta
4	Funkcije za traženje uređaja, mogućnost ispisa programa sa uređaja ili mogućnost upisa programa u uređaj
5	Funkcije za provjeru rada uređaja, monitoring mod prikaza, tablični prikaz programa sa stanjem i adresama

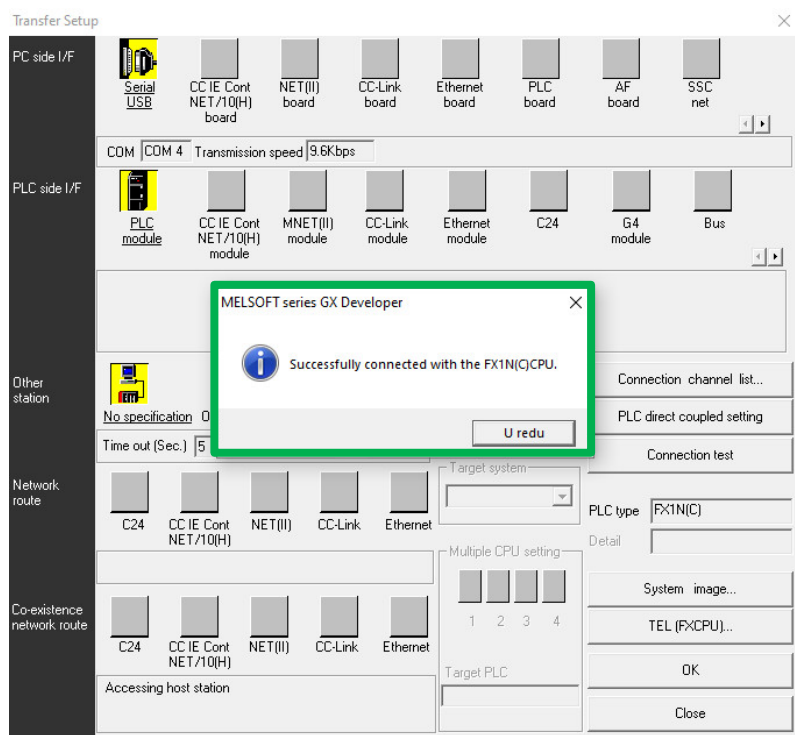
1.10. Povezivanje PLC-a sa računalom

FX1N-32MR se povezuje sa računalom pomoću RS-232C komunikacijskog protokola. Potrebno je prije svega u MELSOFT programu definirati vrstu komunikacije te COM port, također postoji mogućnost odabira brzine.

Nakon toga potrebno je testirati komunikaciju odabirom „Connection test“



Slika 1.12 Prikaz izgleda konfiguracije povezivanja sa FX1N-32MR



Slika 1.13 Prikaz obavijesti uspješnog povezivanja sa FX1N-32MR

1.11. Izrada PLC programa

Izradit će se programska logika za simulaciju rada samoposlužne autopraonice te mogućnošću upravljanjem instaliranim modulom kraj PLC-a na kojem se nalaze tipkala za ručno upravljanje procesa. Također će se naknadno izraditi putem mikrokontrolera ESP32 bežično upravljanje na mogući ručni ili automatski simulacijski rad.

Device name	Comment
X000	VENT.
X001	MODUL UPR.
X002	ISPIRANJE
X003	PRANJE
X004	PRANJE-MOB
X005	ISPIRANJE-MOB
X006	VENT.MOB
X007	START AUTO/RUCNO-MOB

Slika 1.14 Tablični prikaz odabranih broja ulaza PLC-a sa opisom

Device name	Comment
Y000	RAD VENT.
Y001	RAD PUMP
Y002	RAD VENTIL

Slika 1.15 Tablični prikaz odabranih broja izlaza PLC-a sa opisom

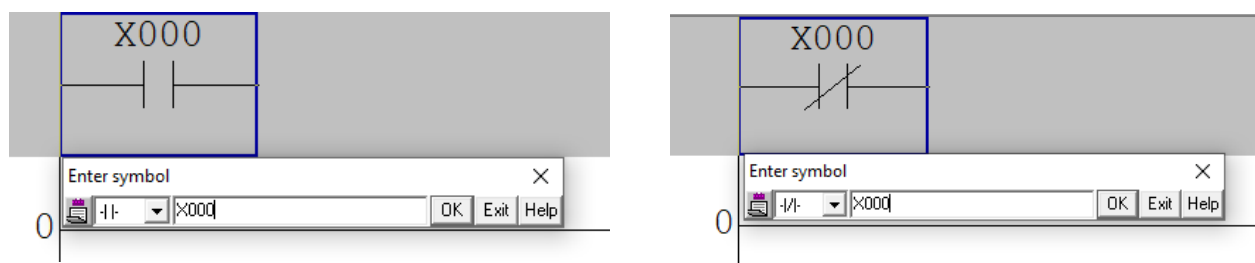
Proces upravljanja samoposlužne autopraonice izrađenim u laboratorijskim uvjetima zamišljen je na način rada u ručnom režimu putem instaliranog modula tipkala kraj PLC-a ili putem korisničkog mobilnog sučelja i na način automatskog rada putem mobilnog sučelja upravljanja.

Ulazni prekidač X0.1 služi za određivanja rada za upravljanje putem instaliranog modula ili mobilnog sučelja, ako je uključeno, moguće je upravljati samo putem modula. Tipkalom X0.0 uključuje se ventilator za sušenje, uključivanjem tipkala X0.2 pokreće se pumpa za ispiranje, a tipkalom X0.3 uključuje se istovremeno pumpa za ispiranje te ventil koji miješa toplu vodu sa koncentriranim šamponom za pranje. Svaki proces traje sveukupno 30s. Isključivanjem prekidača X0.1 onemogućuje se ručni rad putem modula, ali omogućuje se upravljanje putem mobilnog korisničkog sučelja preko ESP32 mikrokontrolera. Uključivanjem tipkala X0.4 za pranje, X0.5 za ispiranje te X0.6 za ventilator omogućuje se rad kao i kod rada putem modula. Uključivanjem tipkala X0.7 pokreće se automatski način rada praonice.

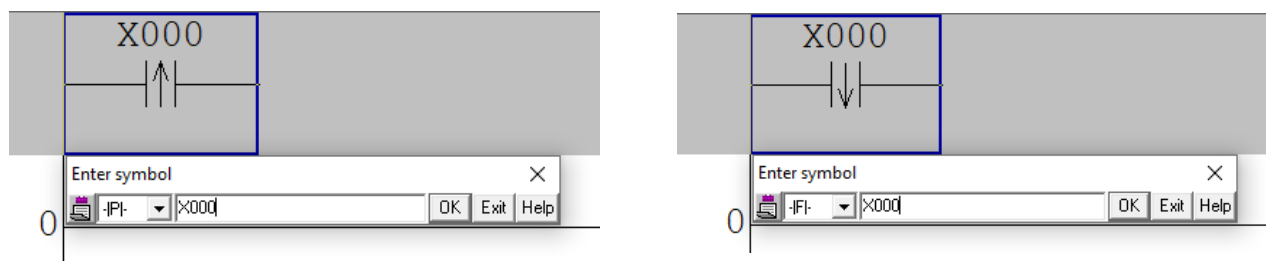
Redoslijed procesa u automatskom režimu rada je sljedeći:

Proces ispiranja > Proces pranja > Proces ispiranja > Proces sušenja > Kraj rada

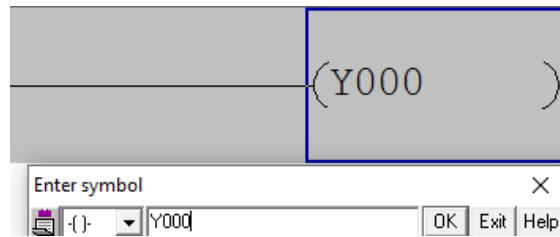
U MESOFT programskom sučelju djelomično je drugačiji princip programiranja PLC-a nego kod STEP 7 programa. Prije samog izrađenog simulacijskog programa objašnjena je logika programiranja u MELSOFT programu.



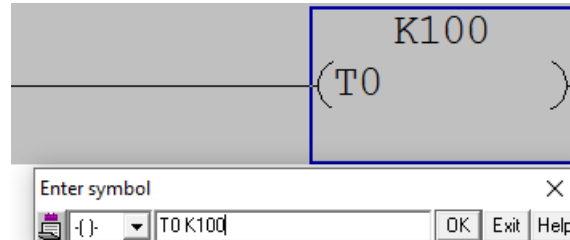
Slika 1.16 Prikaz otvorenog (lijevo) i zatvorenog (desno) kontakta



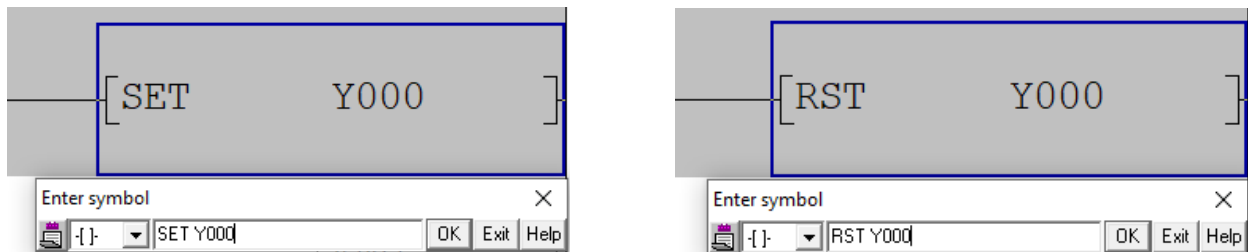
Slika 1.17 Prikaz otvorenog kontakta sa pozitivnim(rastućim) impulsnim bridom (lijevo) i otvorenog kontakta sa negativnim(padajućim) impulsnim bridom (desno)



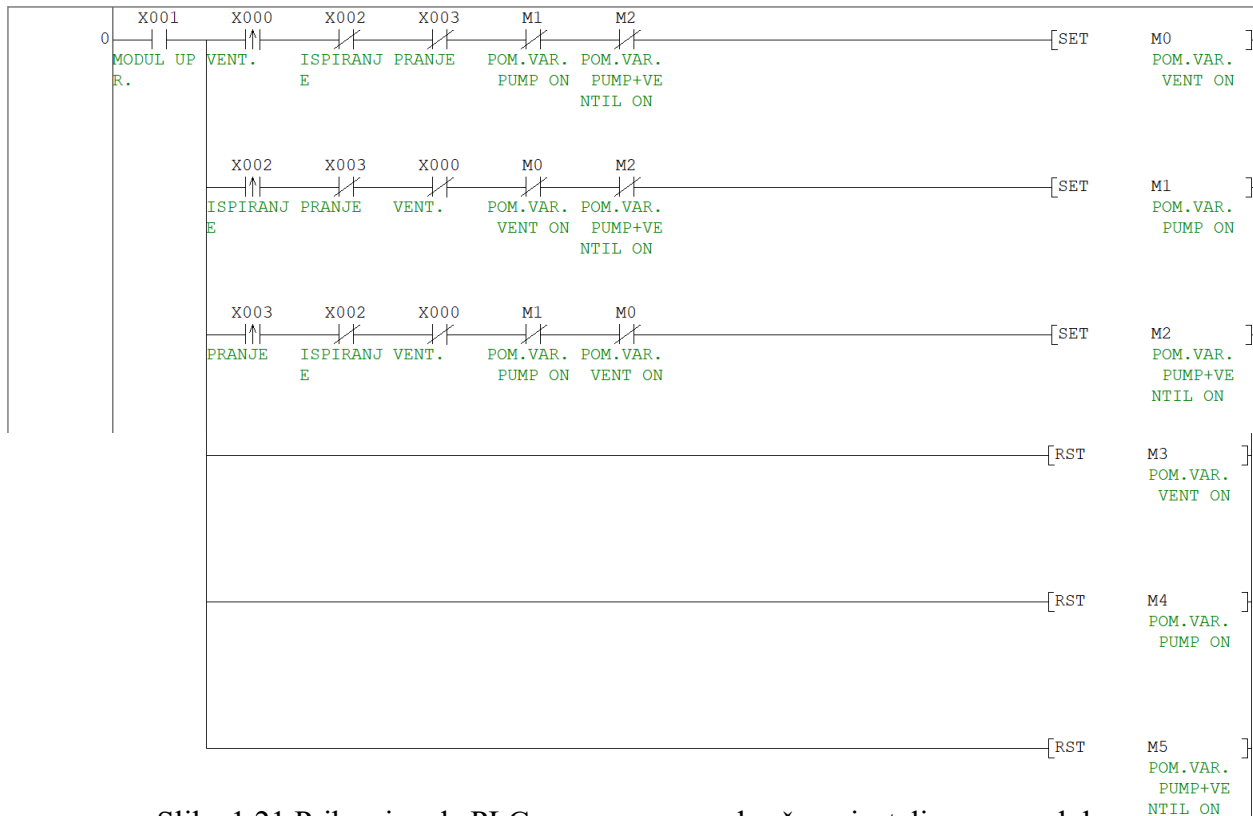
Slika 1.18 Prikaz zavojnice releja – izlaz PLC-a



Slika 1.19 Prikaz definiranja timera T0 – oznaka ; K100 – trajanje od 10s



Slika 1.20 Prikaz definiranja funkcija SET i RESET izlaza Y0.0

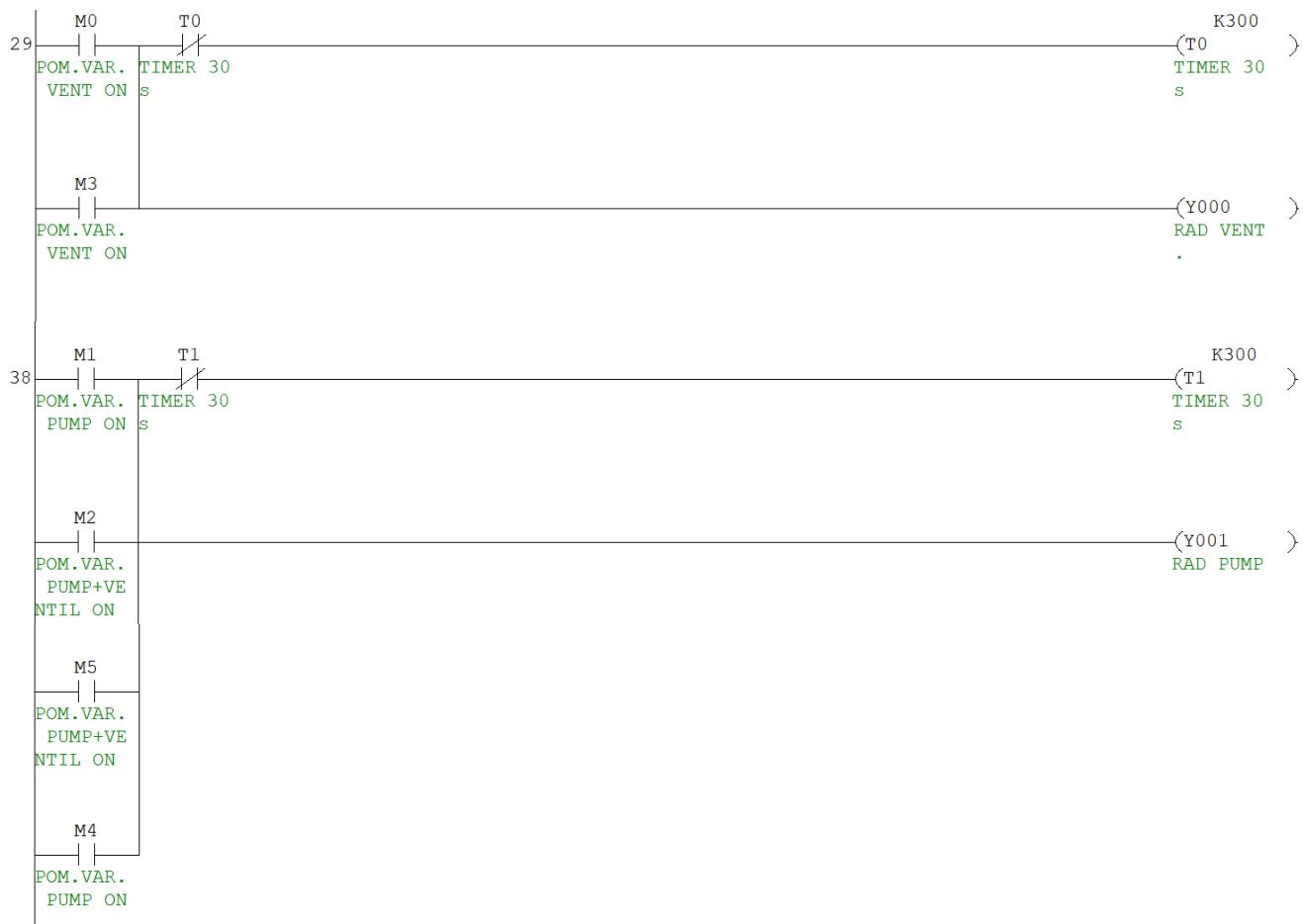


Slika 1.21 Prikaz izrade PLC programa za rad ručnog instaliranog modula

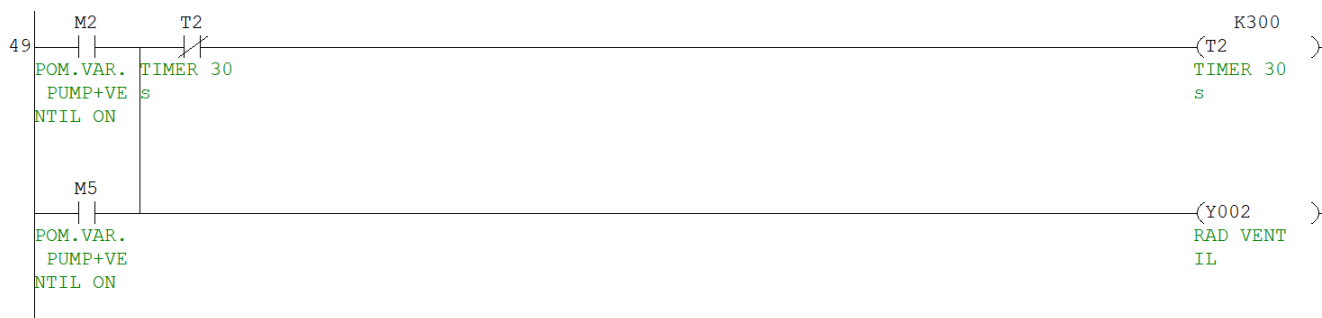
Prema slici 1.21 može se uočiti dio programa za uključivanje ručnog načina rada putem instaliranog modula tipkala kraj PLC-a.

Da bi se lakše odredio rad definiranog procesa u programu PLC-a mogu se koristiti pomoćne varijable za uključivanje dijelova programa unutar određenog procesa. Svaki proces je definiran naredbama **SET** i **RESET** pomoćne varijable **M**.

Prilikom uključjenja neke od pomoćnih varijabli uključuje se timer koji odbrojava vrijednost od 30s gdje pomoćna varijabla uključuje neki izlaz, a kasnije timer resetira pomoćnu varijablu.



Slika 1.22 Prikaz djela programa sa timerom od 30s za uključenje izlaza Y0.0 (rad ventilatora) i Y0.1(rad pumpe)



Slika 1.23 Prikaz djela programa sa timerom od 30s za uključenje izlaza Y0.2 (rad ventila)



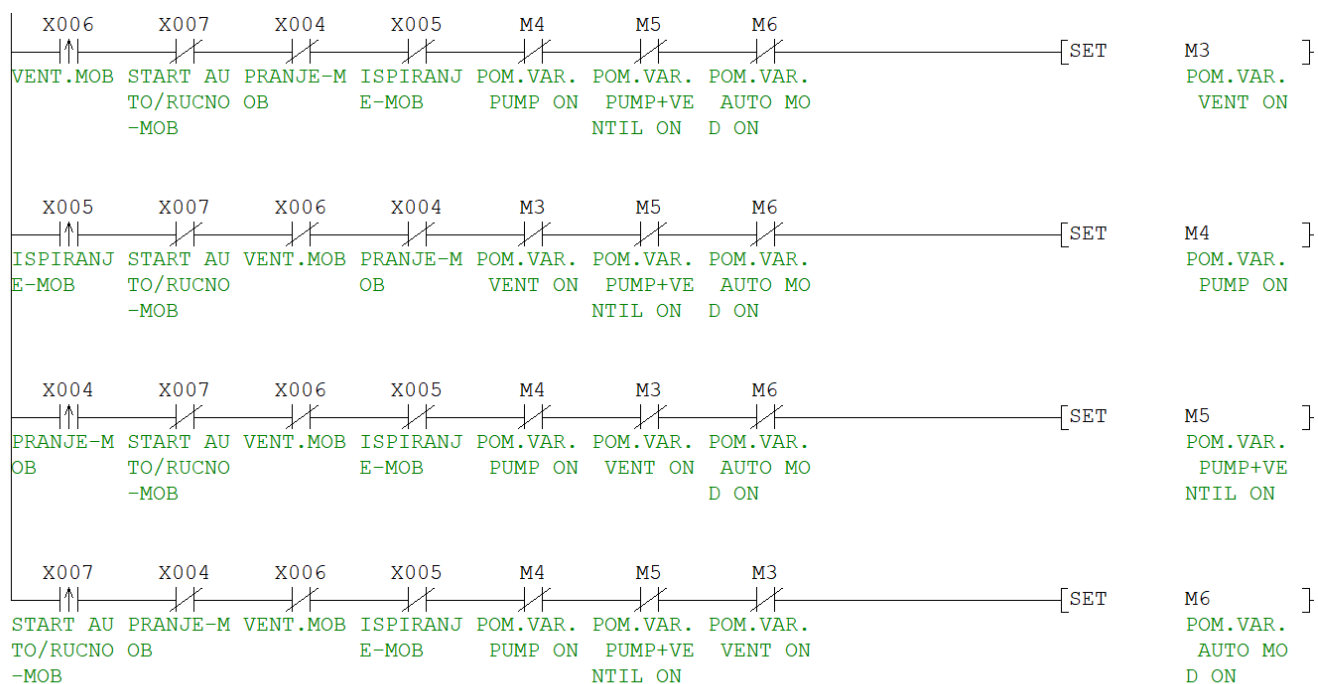
Slika 1.24 Prikaz djela programa za reset pomoćne varijable preko timera

Kada je mogućnost ručnog režima rada isključena, odnosno ulazna sklopka X0.1 je isključena, dolazi do postavljanja svih pomoćnih varijabli ručnog režima rada u **stanje 0**, odnosno resetiranja.



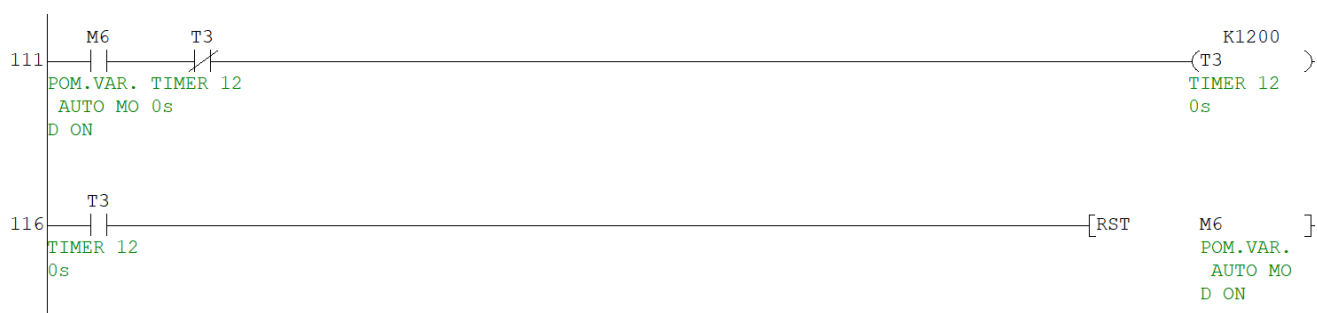
Slika 1.25 Prikaz djela programa kod isključenja ulazne sklopke X0.1

Nakon ručnog rada putem instaliranog modula te isključenjem sklopke X0.1 mora se osigurati da su svi procesi završeni.



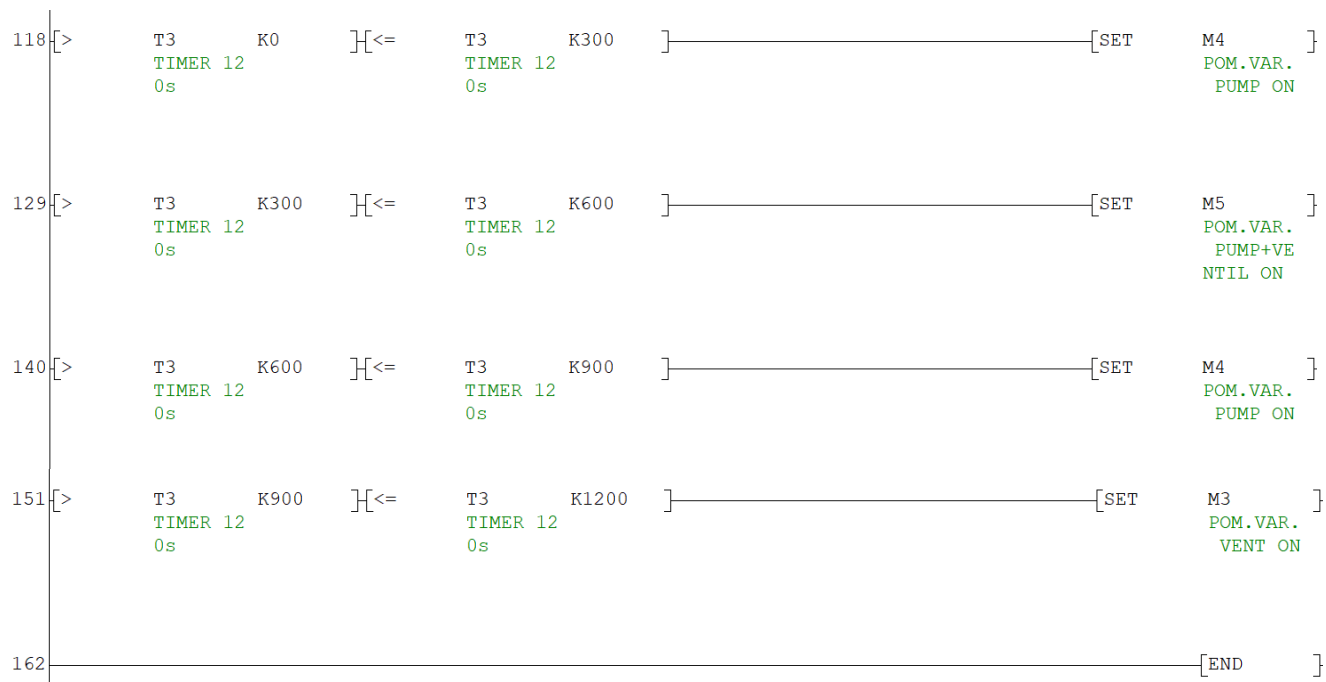
Slika 1.26 Prikaz djela programa kod upravljanja putem mikrokontrolera

Rad putem korisničkog mobilnog sučelja baziran je na komunikaciji putem ESP32 mikrokontrolera. Izlazni pinovi mikrokontrolera indirektno uključuju ulazni pin PLC uređaja. Rad automatskog režima rada definiran je pomoćnom varijablom M6 koja pokreće timer od ukupno 120s trajanja prema navedenom redosljedu procesa.



Slika 1.27 Prikaz djela programa za uključenje timera automatskog režima rada

Svaki proces se poziva unutar vremenskog intervala timera T3 preko svojih pomoćnih varijabli koji uključuju svoj proces.

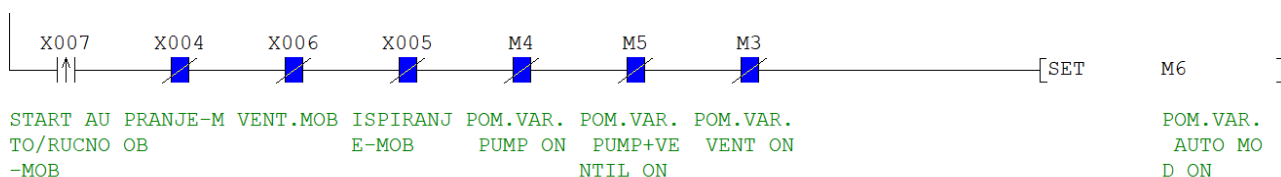


Slika 1.28 Prikaz djela programa za pozivanje rada procesa unutar vremenskog intervala

1.12. Simulacija PLC programa

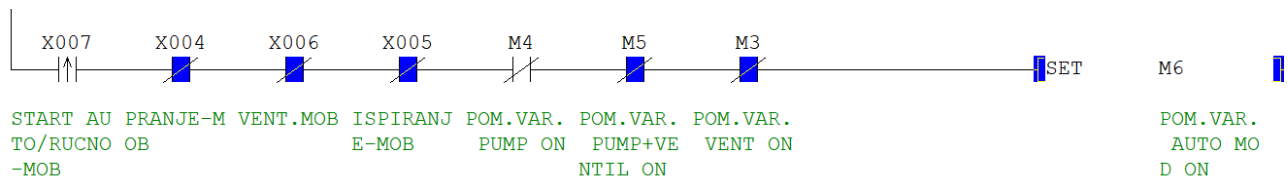
Simulacijom PLC programa u monitoring okruženju može se uspostaviti veza rada programa te predvidjeti eventualne moguće pogreške tijekom promjena procesa.

Kao primjer simulacije pokrenut je automatski režim rada samoposlužne praonice.



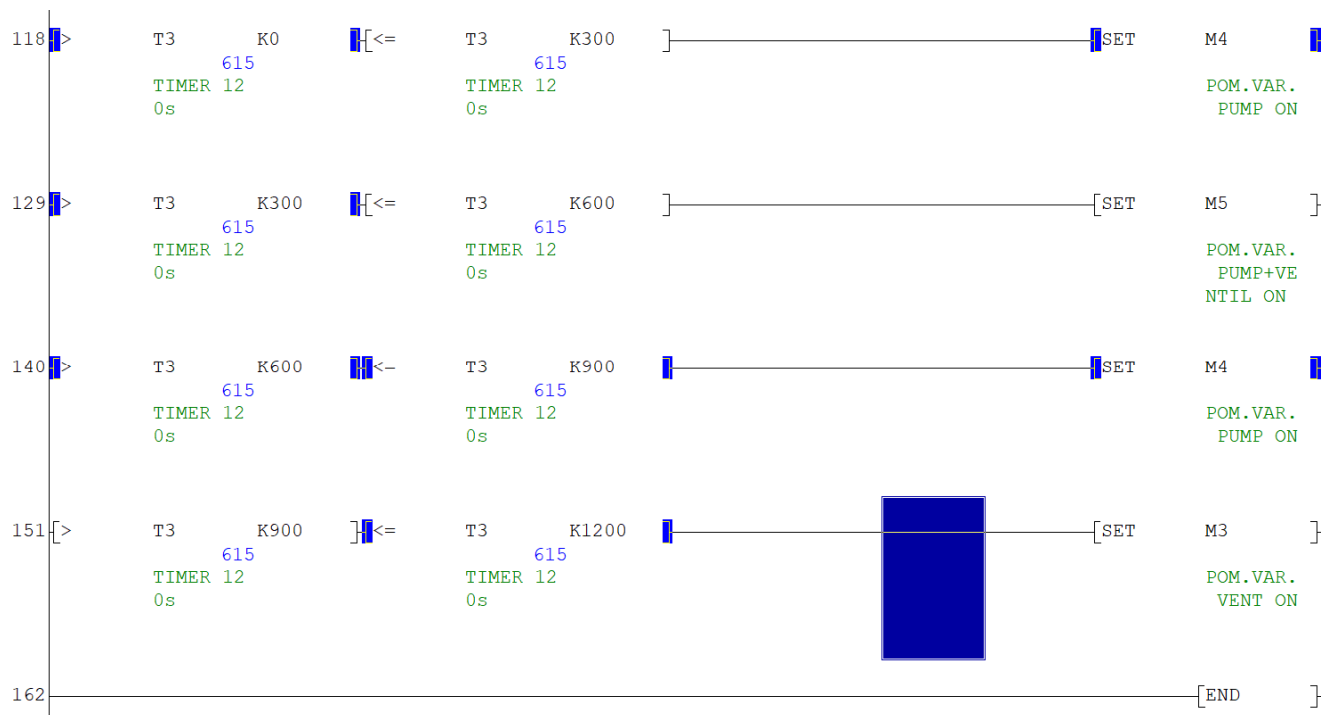
Slika 1.29 Prikaz simulacije programa kod automatskog rada – X0.7 je isključen

Prilikom isključenja tipkala X0.1 moguće je uključiti mobilnim sučeljem automatski rad. Kao što je navedeno u PLC procesu X0.7 je kontakt koji postavi pomoćnu varijablu **M6** na **stanje 1** sve do kraja trajanja timera T3.



Slika 1.30 Prikaz simulacije programa kod automatskog rada – X0.7 je uključen

Kada je X0.7 bio uključen, stanje pomoćne varijable **M6** je postavljeno na **stanje 1**.



Slika 1.31 Prikaz simulacije programa kod odbrojavanja rada timera T3

Prema slici 1.31 moguće je uočiti rad pojedinog procesa unutar određenog vremenskog intervala.

Proces ispiranja odnosno rad pumpe određen je pomoćnom varijablom **M4** koja prema uvjetu se postavlja na **stanje 1**

ako je:

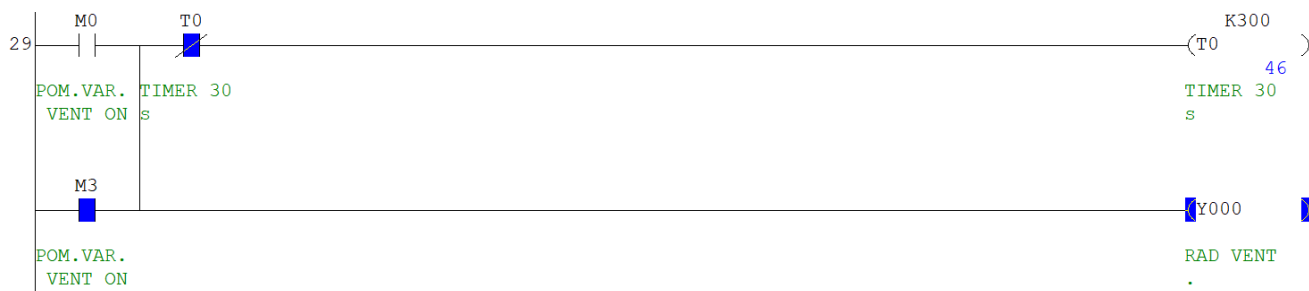
polazno vrijeme > od polaznog vremena timera T3 0s

I

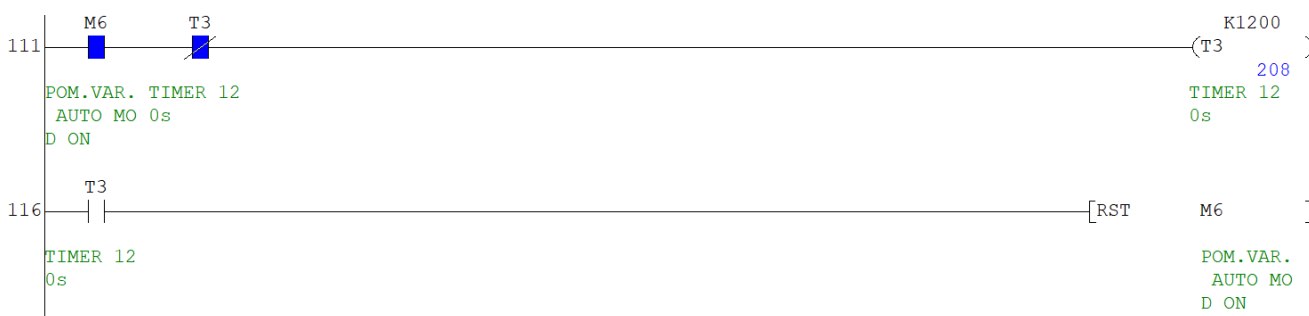
ako je:

polazno vrijeme <= polazno vrijeme od timera T3 30s

Dakle, proces traje od početka te do 30.-te sekunde ukupnog vremena T3.



Slika 1.32 Prikaz simulacije programa kod rada ventilatora Y0.0



Slika 1.33 Prikaz simulacije programa kod rada timera T3

Pomoćna varijabla **M6** se nakon 120s postavlja na **stanje 0** kada se uključuje **timer T3** i time proces automatskog rada završava.

1.13. Prednosti i nedostaci PLC uređaja

Kod takvih automatiziranih sustava važno je da PLC radi u najboljem režimu rada, a to svakako ovisi o puno faktora. Programiranjem PLC-a može se detaljno obratiti pozornost na svaku situaciju sustava kod mogućih kvarova, javljanja grešaka ili kod poteškoća prilikom rada. Najveći nedostatak takvih automatiziranih sustava je što su kvalitetni PLC uređaji sa licencom veoma skupi, najčešće zbog toga dođe do odabira mikrokontrolera. Međutim, s druge strane PLC uređaj je vrlo jednostavan za programirati, velika mu je industrijska primjena zbog toga što je dizajniran prema industrijskim procesima, također skuplji PLC uređaj odlikuje veća pouzdanost i preciznost prilikom rada, ugrađenu podršku za više protokola, moguće su izravne komunikacije sa senzoričkom te u slučaju modularnog tipa PLC-a moguće je i proširenje.

2. SCADA – Supervisory Control and Data Acquisition

2.1. Općenito

SCADA (eng. *Supervisory Control and Data Acquisition*) je zapravo računalni sustav za nadzor, mjerenje te upravljanje industrijskim procesima.

SCADA sustav omogućuje kontrolu nad industrijskim procesima u lokalnom ili udaljenom području uz nadzor, prikupljanje i obradu podataka u realnom vremenu sa mogućom izravnom interakcijom s uređajima poput senzora, ventila, pumpi, el. motora i slično. SCADA sustavi su presudni u industrijama jer pomažu u održavanju učinkovitosti uz obradu podataka i donošenjem „pametnih“ odluka putem komunikacije kod nastalih problema kako bi se ublažilo ili gotovo anulirao zastoj procesa.

2.2. Povijest i razvoj SCADA sustava

Prve izvedbe SCADA sustava započeli su samostalno na računalu bez ikakvih komunikacija jer u to doba mrežne tehnologije nisu bile dostupne ni razvijene kao danas. Takvi SCADA sustavi bi se danas nazivali „monolitnim“ SCADA sustavima. U 80-ima i 90-ima godinama SCADA se nastavila razvijati kod manjih računalnih sustava pomoću lokalnih mreža (LAN) i HMI (eng. Human-Machine Interface) softveru koji je temelj na računalu. Tim razvojem SCADA sustavi su se mogli povezati s drugim sličnim sustavima, uz korištenjem raznih LAN protokola koji su bili dio vlasništva neke industrije nije bilo baš moguće komunicirati sustavima drugih industrija zbog nekompatibilnosti te vlastite sigurnosti industrije, takvi sustavi su se nazivali distributivni SCADA sustavi.

1990-ima i ranim 2000-ima SCADA sustavi su se postupno počeli mijenjati te su privatni sektori (industrije) počele prihvaćati otvorenu arhitekturu sustava i komunikacijske protokole time se dovelo do razvoja umrežavanja SCADA sustava što je omogućavalo kvalitetniju međusobnu komunikaciju uz blaga ograničenja.



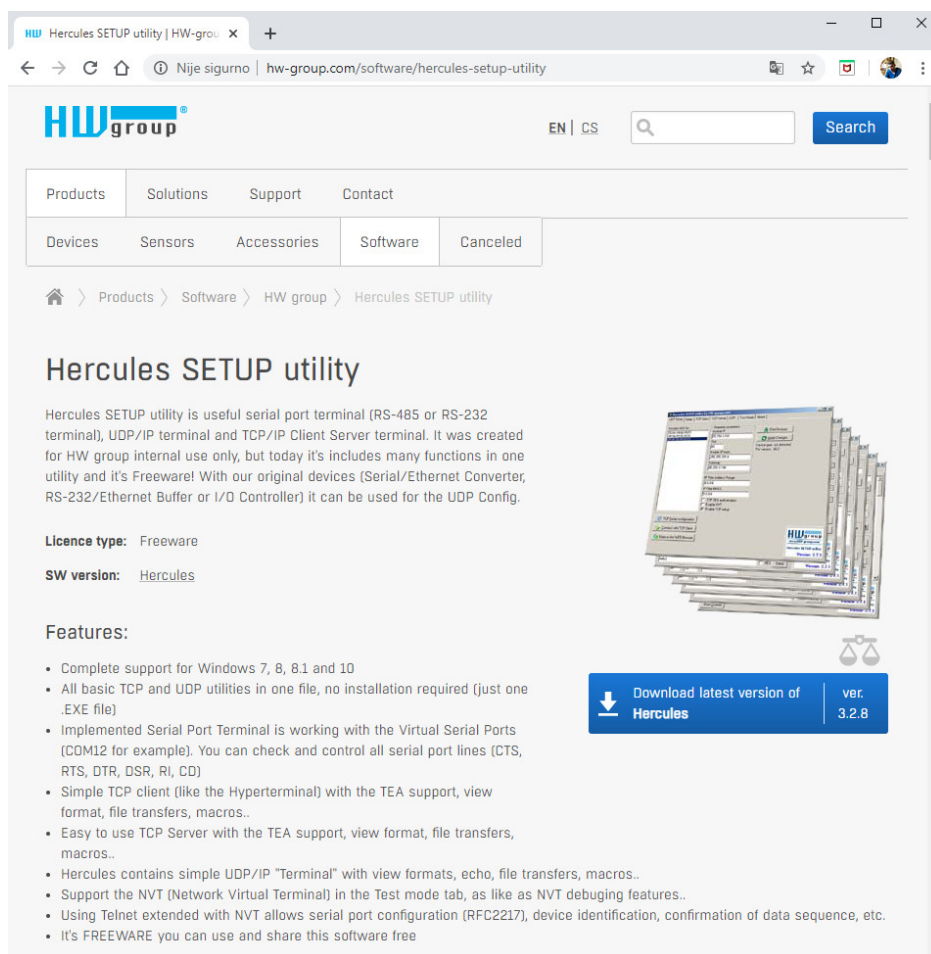
Slika 2.1 Prikaz prvih izgleda „monolitnih“ SCADA sustava [8]

2.3. Komunikacija PLC-a i SCADA sustava

Kao primjer razumijevanja SCADA sustava vizualizacije procesa prije svega je potrebno razumijevanje komuniciranja PLC uređaja sa računalom, odnosno programskim paketom vizualizacije u ovome slučaju izrađenim u SIMATIC WinCC-u. Realizacija vizualizacije će se kasnije opisati, međutim prije toga potrebno je definirati bitne stavke kod povezivanja FX1N-32MR PLC uređaja sa SIMATIC WinCC programom, a to je vrsta komunikacije.

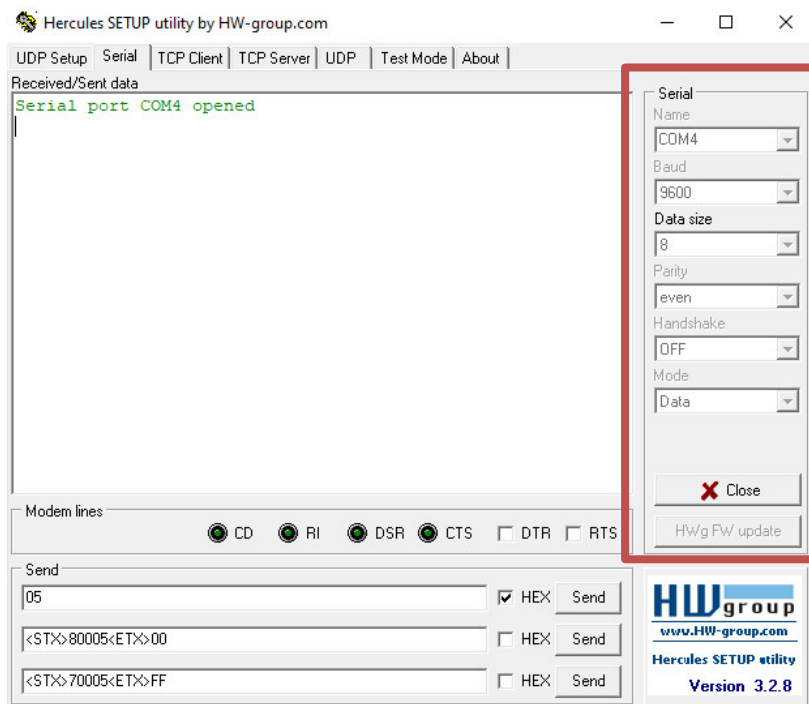
SIMATIC WinCC programski paket služi za izradu SCADA sustava sa svim potrebnim stavkama procesa koje SCADA sustav može pružati. U radu je izrađena vizualizacija procesa sa direktnim upravljanjem na PLC uređaj.

Prije svega, da se osigura ispravnost rada RS-232 serijske komunikacije putem **Hercules SETUP**, terminal serijskog priključka za komunikaciju (RS-485 ili RS-232), UDP, IP te TCP/IP klijentski poslužitelj, će se ispitati funkcionalnost komunikacije. To je besplatan program s mnogo funkcija. Koristit će se u ovom slučaju samo za provjeru serijskog priključka.



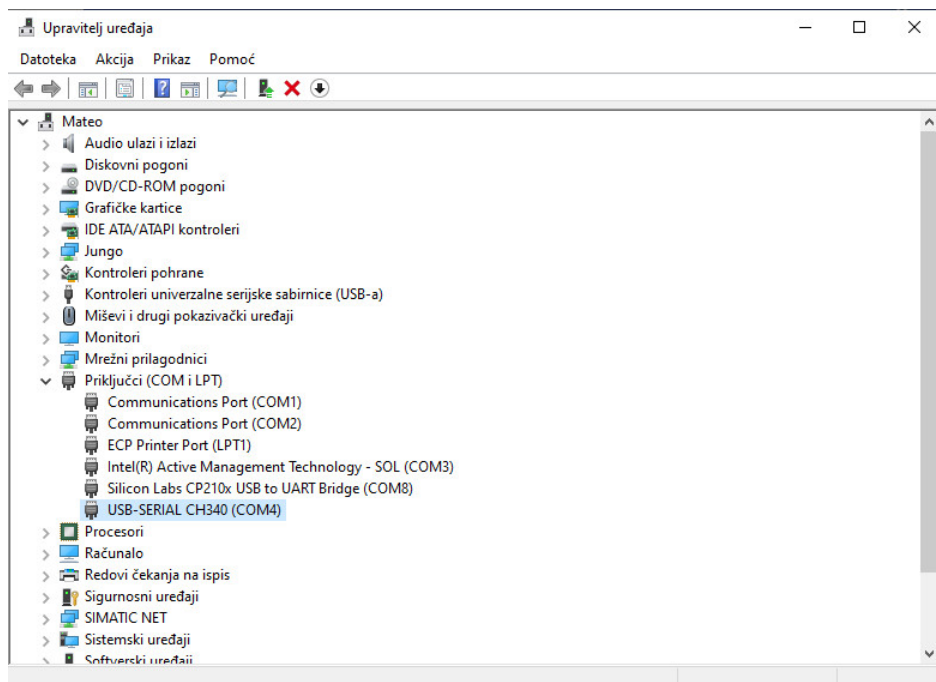
Slika 2.2 Prikaz hw-group stranice za preuzimanje terminala Hercules [9]

Prilikom ulaska u program koristi se kartica **Serial** i time se definira s desne strane COM port priključak, brzina, veličina podataka te model **Data** kao vrsta prijenosa podataka.



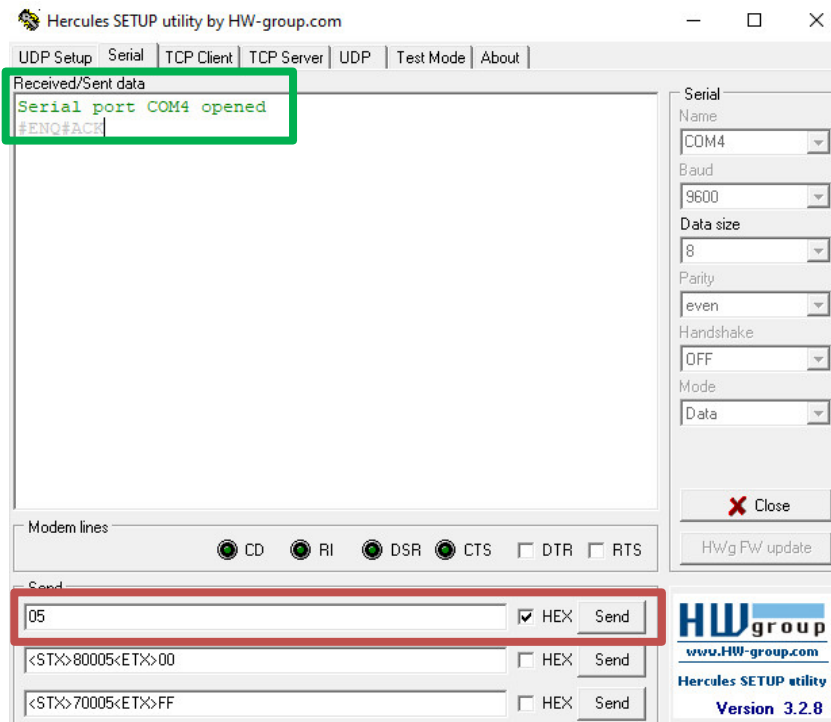
Slika 2.3 Prikaz Hercules terminala u okruženju serijske komunikacije

Nakon toga uključuje se port **COM4**. Provjeru točnog porta može se provjeriti kod upravitelja uređaja.



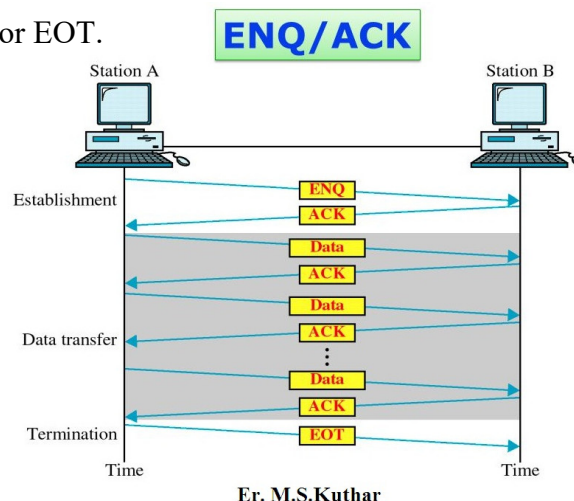
Slika 2.4 Prikaz COM porta kod upravitelja uređaja

Nakon što se port otvori, komunikacijski kanal provjerava se jednostavnom metodom slanja obične HEX adrese **05**. Prilikom slanja adrese dobiva se potvrda **#ENQ**, a prilikom zaprimanja adrese dobiva se potvrda **#ACK**. Time se zaključuje da je komunikacija ispravna te prijenos podataka radi.



Slika 2.5 Prikaz Hercules terminala kod slanja HEX adrese 05

U pravilu inicijator prvo prenosi okvir koji se naziva upitom ENQ, pitajući je li prijammnik dostupan za primanje podataka. Primatelj mora odgovoriti okvirom potvrde ACK ako je spreman za prijam ili okvirom negativnog odgovora NAK ako nije. Ako u tom slučaju nije zaprimljen ni jedan od tih odgovora, inicijator pretpostavlja da je ENQ okvir izgubljen u prijenosu, prekidajući slanje te uobičajeno ponovi postupak sve zajedno tri puta prilikom uspostave veze. Ako je odgovor negativan na sva tri pokušaja, inicijator prekida proces. Prilikom završavanja slanja podataka okvirom dobiva se odgovor EOT.

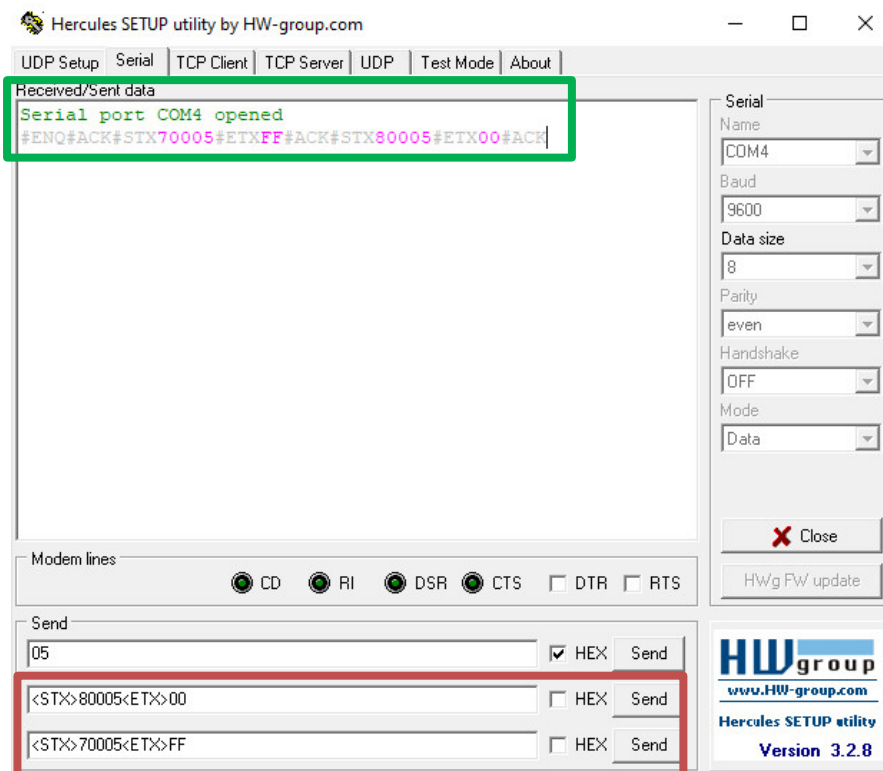


Slika 2.6 Prikaz odnosa okvira ENQ/ACK [10]

STX (Start-of-Text character) je kodni karakter koji se koristi za obavještanje računala, točnije primatelja da je dosegnut početak zapisa, a ETX (End-of-Text character) da je dosegnut kraj zapisa.

<STX>80005<ETX>00

<STX>70005<ETX>FF



Slika 2.7 Prikaz uključivanja i isključivanja izlaza PLC-a Y0.0 putem terminala

Kodnim karakterima STX te ETX može se definirati adresom točan ulaz ili izlaz PLC-a te odrediti njegovo stanje (1,0). U primjeru 80005 je adresa izlaza Y0.0 kojom je moguće samo izlaz dovesti u nisko stanje, adresom 00, odnosno 70005 je adresa izlaza Y0.0 kojom je moguće samo izlaz dovesti u visoko stanje, adresom FF. Prilikom svakog završenog prijenosa dobiva se potvrdni odgovor okvirom ACK.



ASCII Table

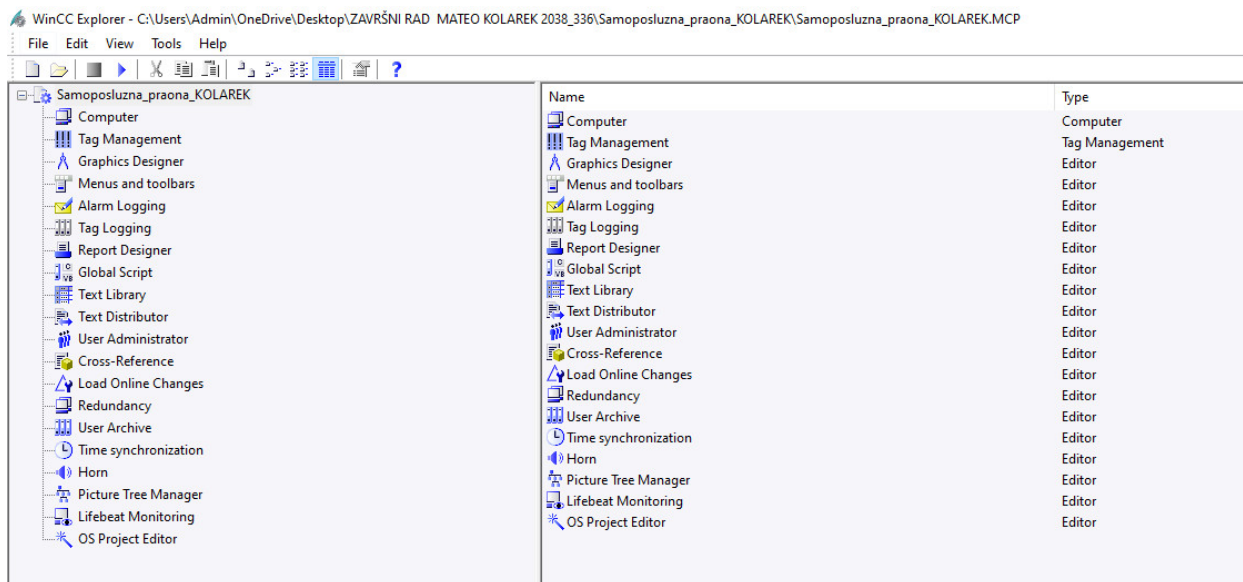
Non-Printing Characters				Printing Characters									
Name	Ctrl char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
null	ctrl-@	0	00	NUL	32	20	Space	64	40	@	96	60	`
start of heading	ctrl-A	1	01	SOH	33	21	!	65	41	A	97	61	a
start of text	ctrl-B	2	02	STX	34	22	"	66	42	B	98	62	b
end of text	ctrl-C	3	03	ETX	35	23	#	67	43	C	99	63	c
end of xmit	ctrl-D	4	04	EOT	36	24	\$	68	44	D	100	64	d
enquiry	ctrl-E	5	05	ENQ	37	25	%	69	45	E	101	65	e
acknowledge	ctrl-F	6	06	ACK	38	26	&	70	46	F	102	66	f
bell	ctrl-G	7	07	BEL	39	27	'	71	47	G	103	67	g

Slika 2.8 Prikaz STX i ETX iz ASCII tablice [11]

2.4. WinCC Explorer

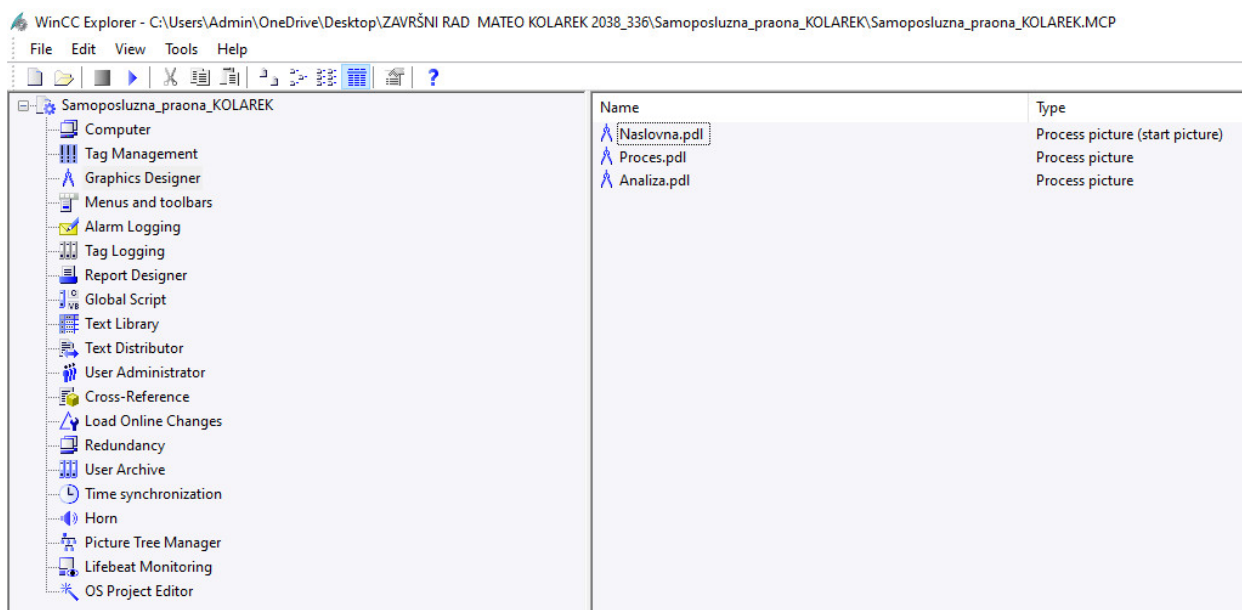
Izradom vizualizacije definiranog procesa PLC programa može se upotpuniti izgled i upravljanje samim procesom. Pokretanjem programa SIMATIC WinCC Explorer pokreće se novi projekat pod imenom „Samoposlužna_praona_KOLAREK“ u kojem će se vizualizirati proces samoposlužne praonice.

Unatoč izgledu proizvođača PLC-a te programskog paketa, povezivanjem Mitsubishi PLC uređaja i SIMATIC WinCC programskog paketa može se omogućiti putem MX OPC servera u kojem se zasebno definira tag lista ulazno-izlaznih pinova PLC-a.



Slika 2.9 Prikaz novog projekta u programu SIMATIC WinCC Explorer

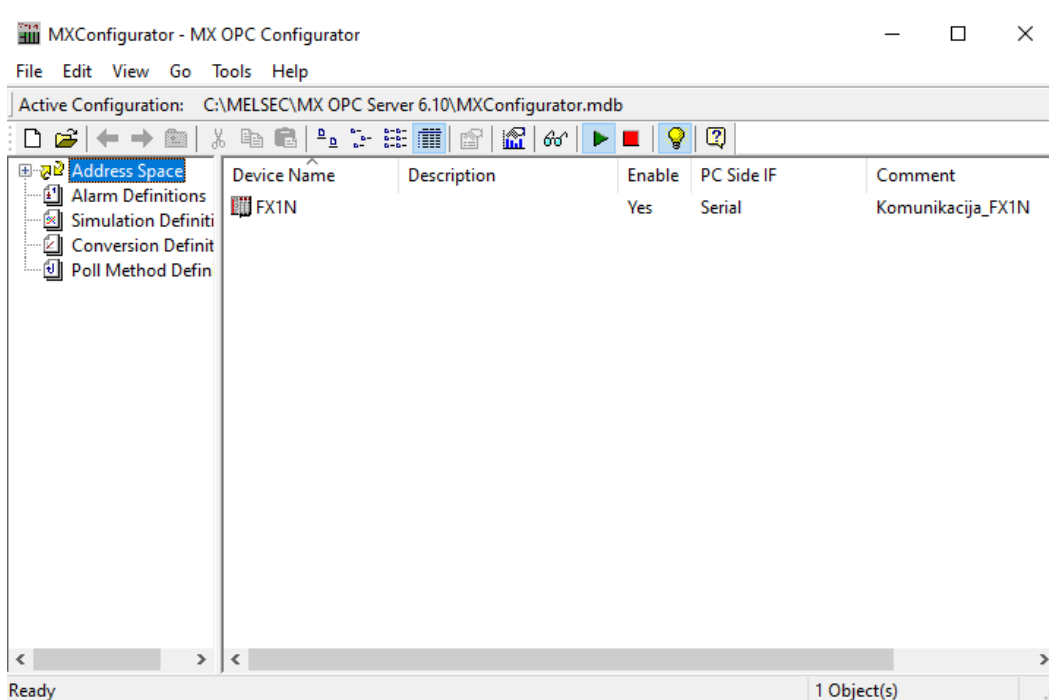
U WinCC programskom paketu započet je projekt izrade vizualizacije, na način se u „Graphics Designer-u“ naprave tri slike, a to su: Naslovna, proces i analiza, a kasnije još sliku samog procesa. Kod naslovne fotografije će se nalaziti opći podaci procesa kao što su naslov, odabir gumba za daljnje slike te izlazna mogućnost. Kod procesa će se napraviti vizualna slika koja će imati mogućnosti prikazivanja trenutnog stanja nekog procesa kada će biti aktivan odnosno neaktivan. Kod analize se može tabličnim ili grafičkim prikazom prikazati opće stanje varijabli, da li su uključene ili ne.



Slika 2.10 Prikaz izrađenih slika vizualizacije

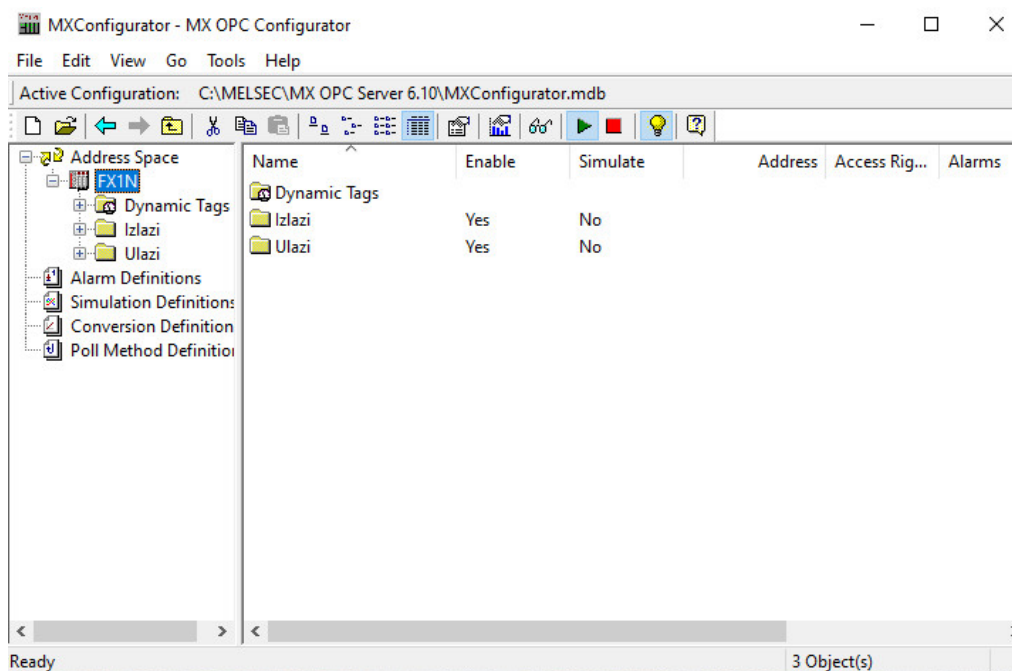
2.5. MX OPC Configurator – server

Da bi se FX1N-32MR PLC uređaj povezo sa SIMATIC WinCC programom koristit će se MX OPC (eng. Mitsubishi *OLE (Object Linking and Embedding) Process Control Configurator*) server. MX OPC server je konfigurator tagova za Mitsubishi tip PLC uređaja koji može preko OPC (eng. *OLE (Object Linking and Embedding) Process Control*) drivera u WinCC-u povezati njegove tagove u svoj „Tag Management“



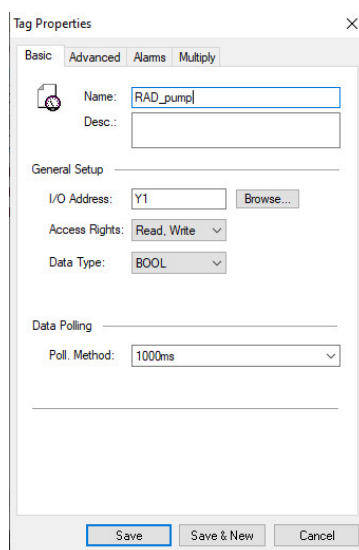
Slika 2.11 Prikaz definiranog PLC uređaja FX1N u MX OPC serveru

Prilikom definiranja Mitsubishi PLC uređaja potrebno je proći čarobnjak kojim se definira način komunikacije, port, brzina prijenosa podataka te vrsta procesora PLC-a. Nakon uspješne izrade mogu se izraditi tagovi. Kao prema slici 1.14 te slici 1.15 može se uočiti tablica sa ulaznim i izlaznim parametrima koji će se unutar MX OPC Configuratora definirati.

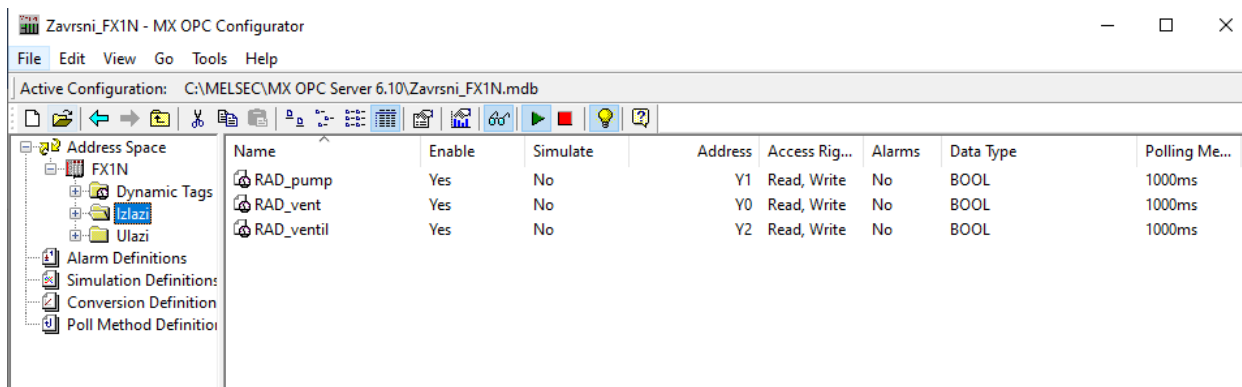


Slika 2.12 Prikaz napravljenih grupa tagova u MX OPC serveru FX1N PLC-a

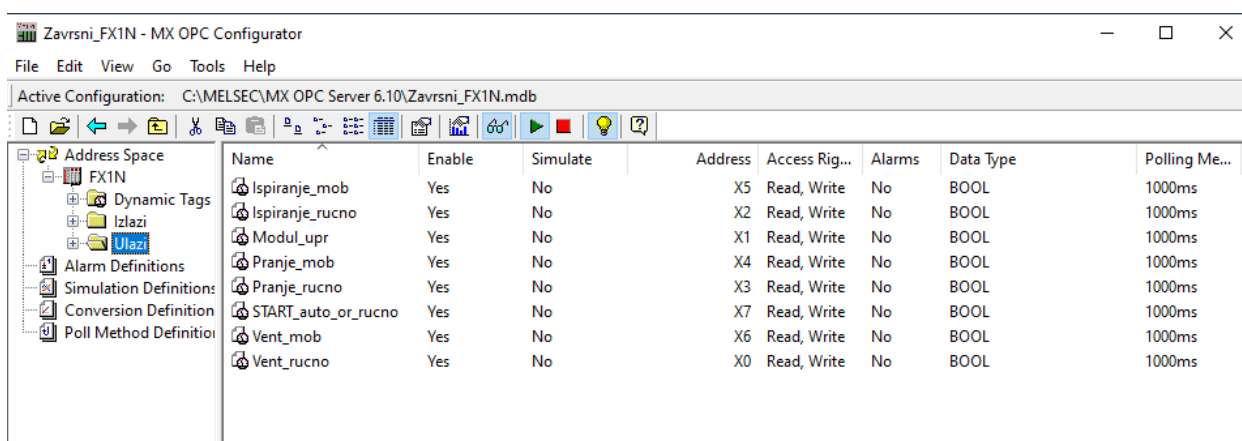
Unutar mape „Izlazi“ se nalaze definirani tagovi izlaznih varijabli PLC-a, a unutar mape „Ulazi“ definirani tagovi ulaznih varijabli PLC-a. Na sljedećoj slici je primjer kako definirati tag unutar MX OPC servera.



Slika 2.13 Prikaz postavke tag-a kod definiranja



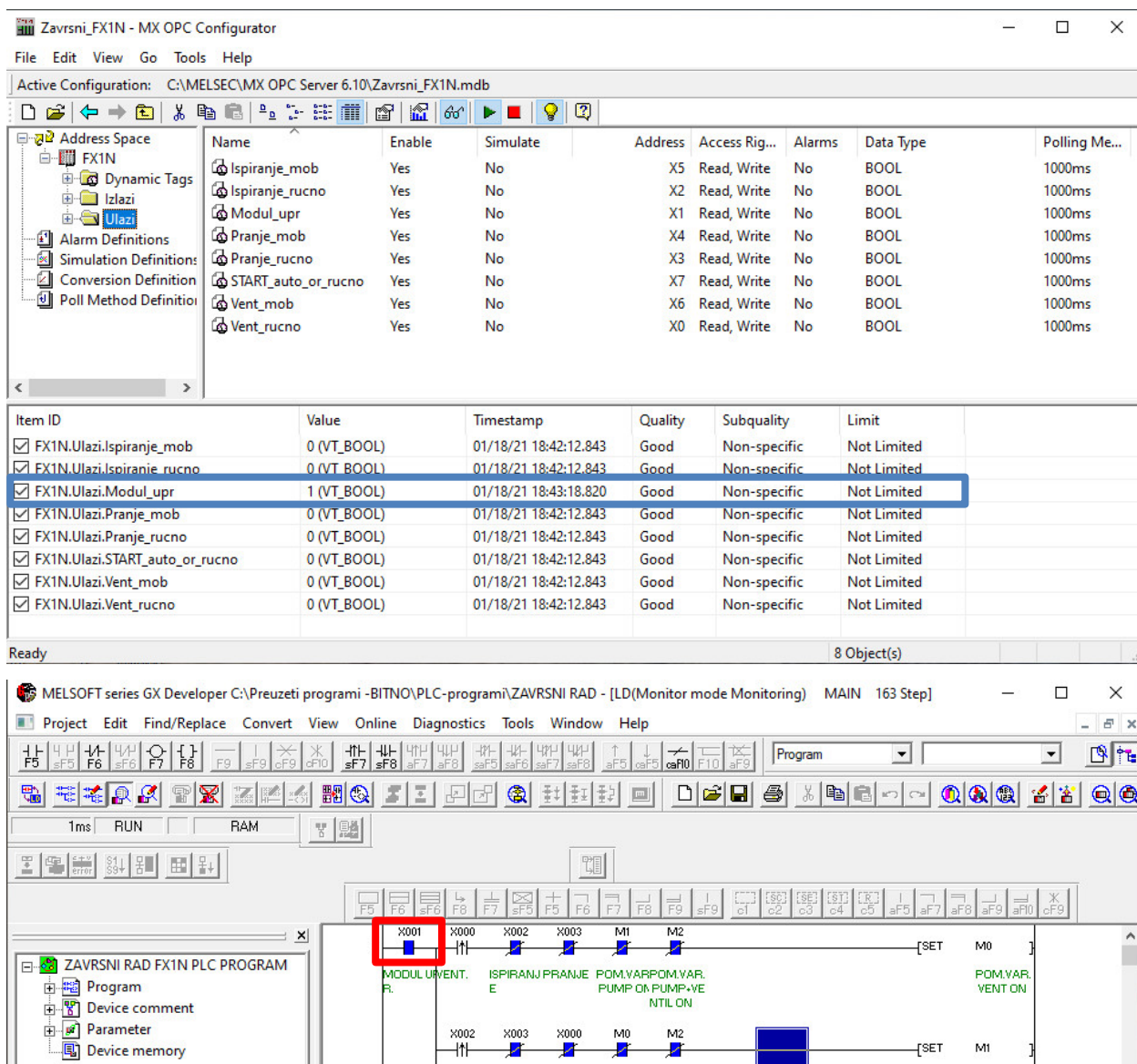
a)



b)

Slika 2.14 Prikaz definiranih tag-ova a) izlaza i b) ulaza FX1N PLC-a

Da bi se provjerio rad OPC servera sa PLC-om uključit će se simulacija PLC-a putem programa te OPC konfiguracija, gdje se mogu vidjeti promjene na definiranim tagovima.



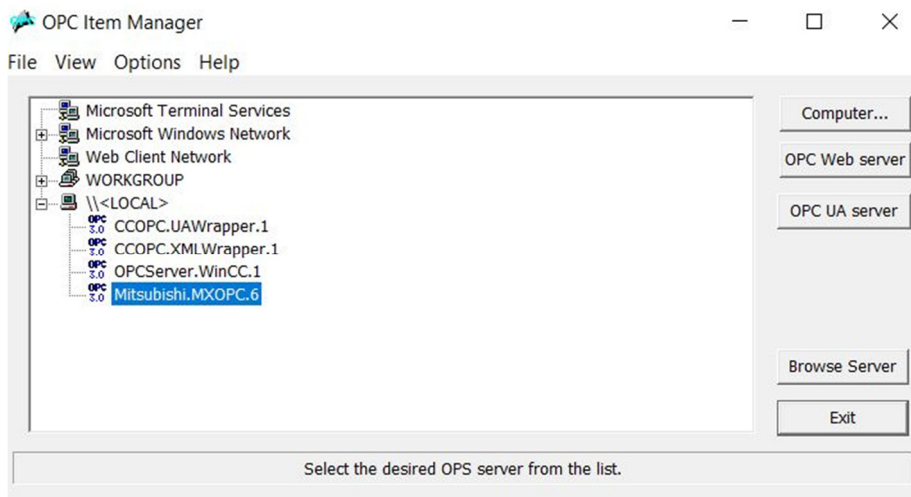
Slika 2.15 Prikaz simulacije OPC servera te kompatibilnost sa programom PLC-a

Prema gornjoj slici kod simulacije OPC servera bitno je polje kvalitete koja mora biti označeno sa **Good**, odnosno kvalitetom da je PLC povezan sa serverom, ako nije, potrebno je provjeriti čarobnjak odabira CPU PLC-a te način povezivanja.

Plava označena boja prikazuje red X0.1 ulaza koji uključuje upravljački modul za ručni rad, kao što se može uočiti crvenom bojom u monitoring radu MELSOFT programa X0.1 je uključen i time je stanje u OPC serveru **1** – što znači da OPC server učitava podatke PLC uređaja.

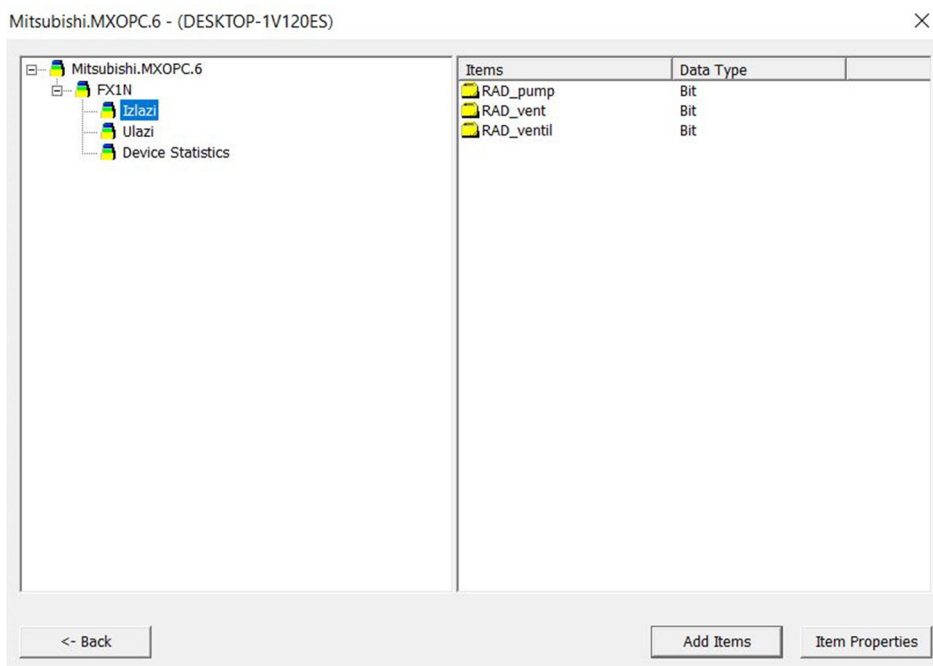
2.6. Povezivanje WinCC-a sa PLC uređajem FX1N-32MR

Nakon definiranja tagova putem MX OPC servera potrebno je u „Tag Management-u“ definirati OPC driver kod kojeg se pronade MX OPC server te se poveže sa pripremljenim tagovima.

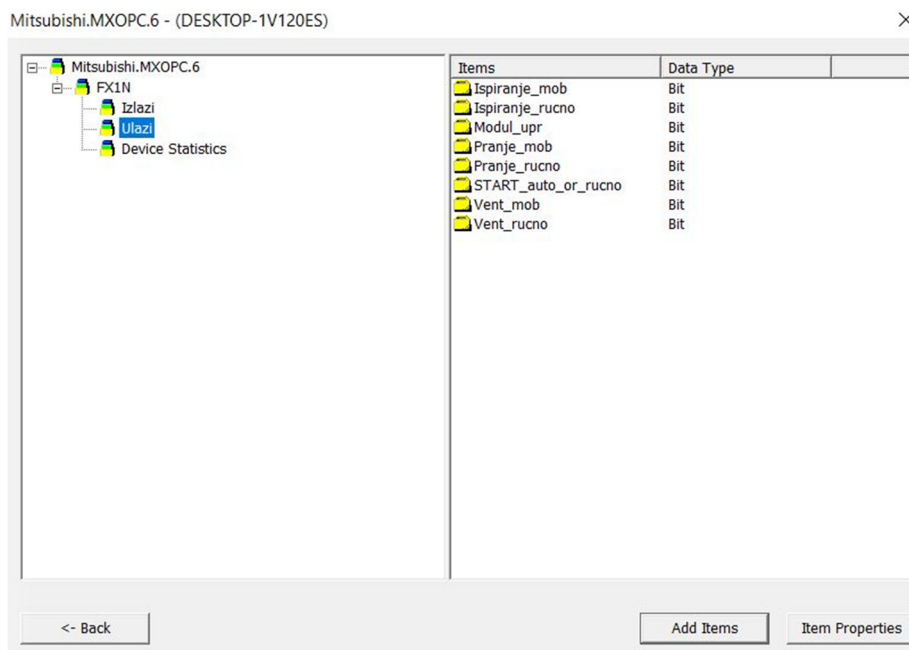


Slika 2.16 Prikaz konfiguratora OPC drivera u WinCC-u

Prilikom pretraživanja MX OPC servera mogu se pronaći definirani tagovi unutar MX Configuratora i time ih se može dodati u WinCC tag listu.

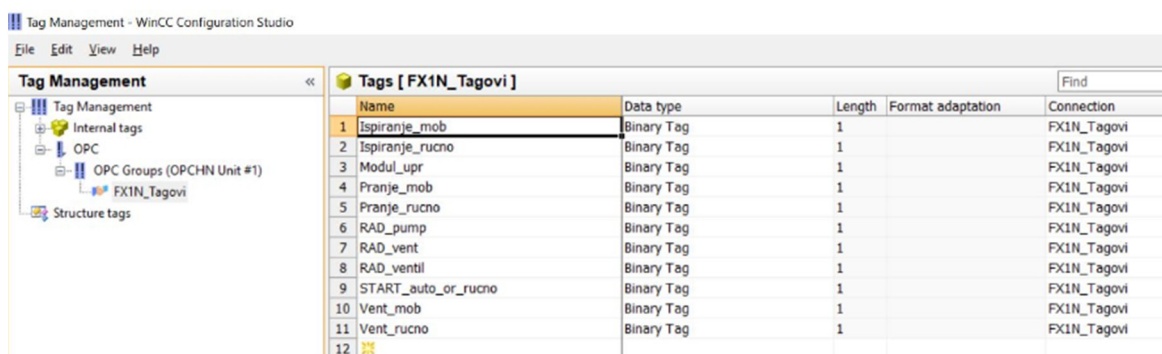


Slika 2.17 Prikaz izlaznih varijabli tagova unutar pretraživača OPC servera



Slika 2.18 Prikaz ulaznih varijabli tagova unutar pretraživača OPC servera

Prilikom dodavanja ulaznih i izlaznih tagova napravljena je podmapa FX1N_Tagovi u kojem su smješteni. Unutar toga mogući je pregled definiranog stanja za svaki tag.

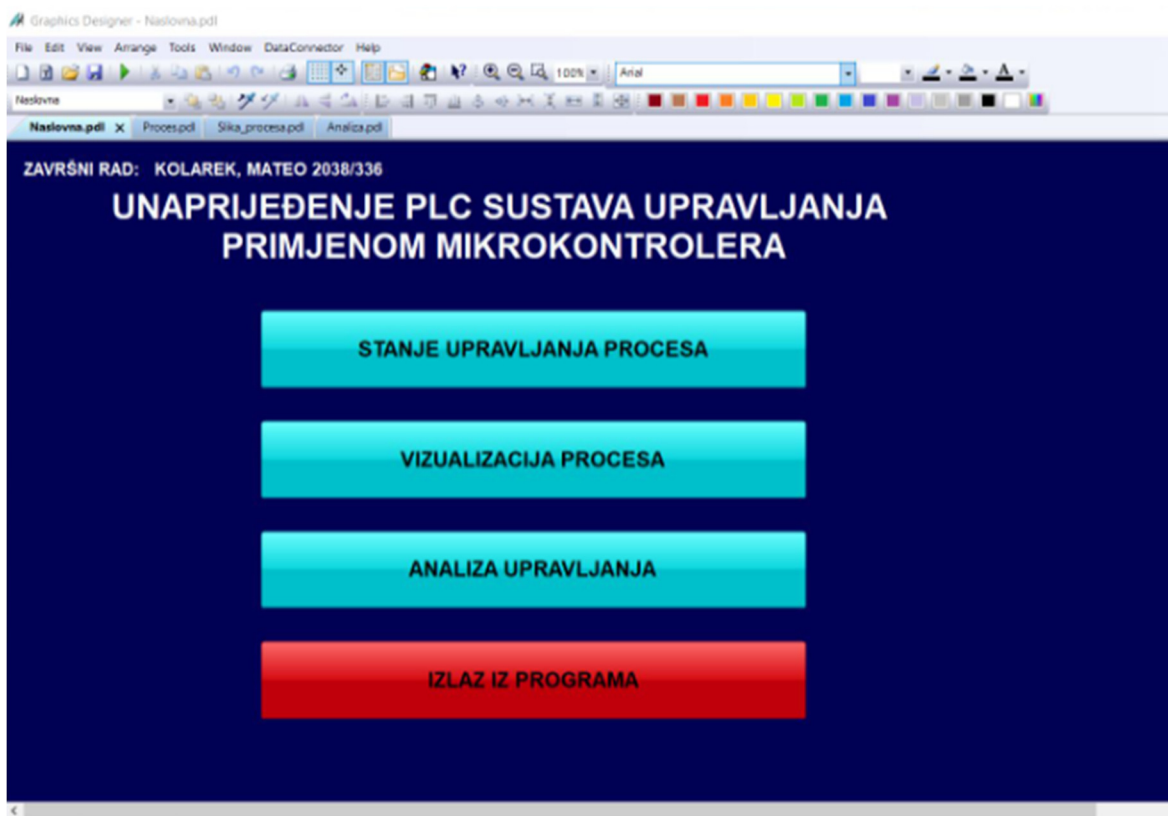


Slika 2.19 Prikaz FX1N tag liste unutar WinCC-a

2.7. Graphics Designer

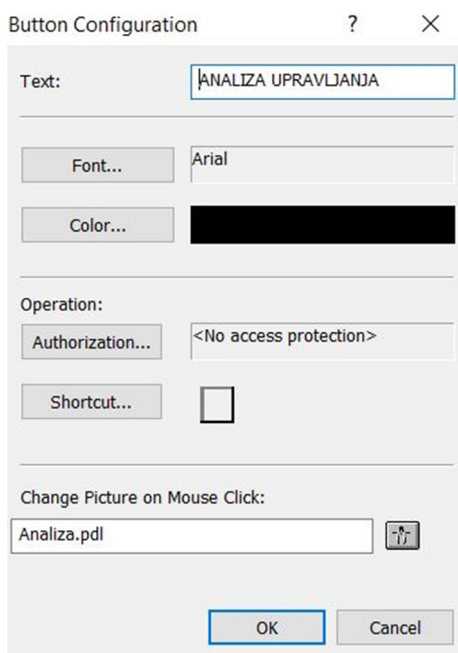
Kada je definirana tag lista, odnosno povezan PLC uređaj sa WinCC programskim paketom, može se krenuti u opis izrade vizualizacije u Graphics Designer-u.

Na naslovnoj fotografiji će se nalaziti naslov rada, mogućnost odabira slike (Proces, analiza te slika samog procesa) i mogućnost izlaza.



Slika 2.20 Prikaz izrade naslovne slike vizualizacije WinCC-a

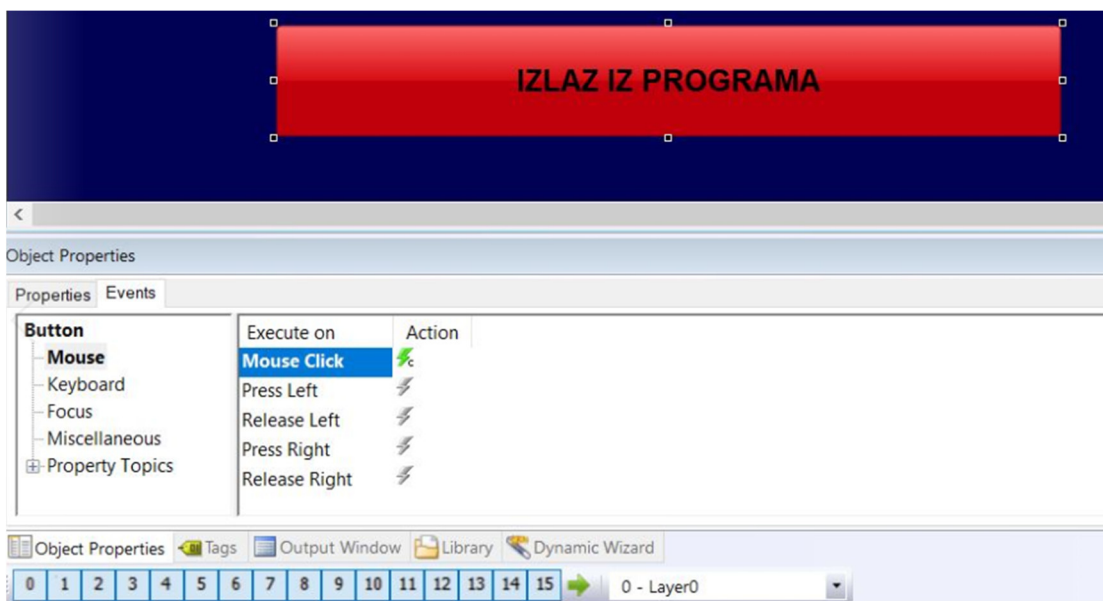
Tipku se može definirati kod standardnih opcija Windows objektima. Prilikom konfiguracije virtualne tipke omogućen je upis naziva tipke, osnovna podešenja teksta te mogućnost odabira dinamike prilikom otvaranja, odnosno akcije na klik. Tom opcijom prve tri tipke se definiraju da otvore na klik priloženu stranicu procesa ili analize.



Slika 2.21 Prikaz konfiguracijskog prozora za definiranje tipke

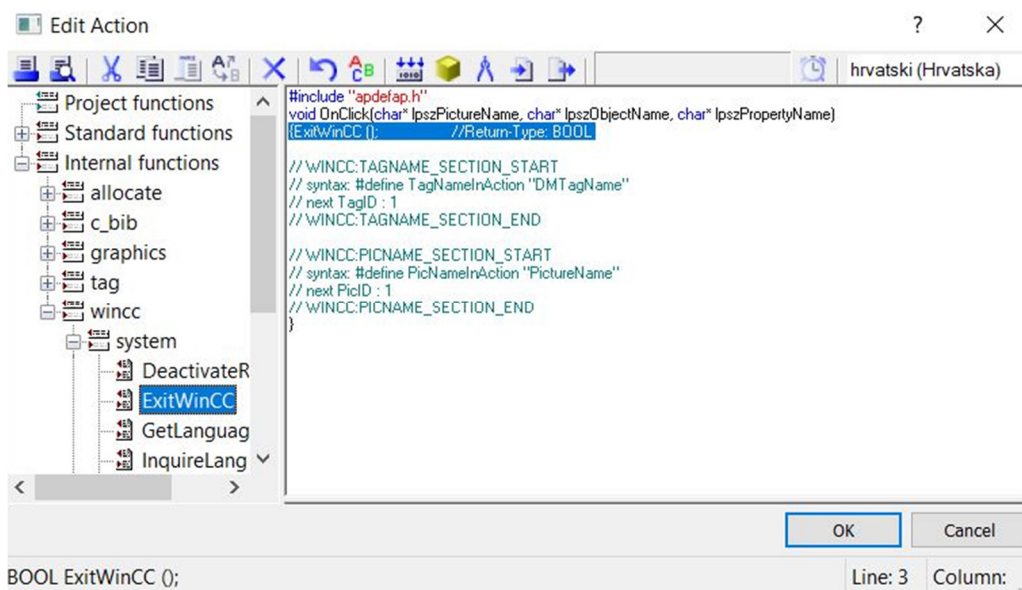
Izlaz iz programa se definira postavkama događaja. U postavkama objekta nalaze se opće postavke te događaji. U događajima postoji mogućnost definirati događaj, odnosno akciju izvršenu klikom, pritiskom na određenu tipku tipkovnice te slično.

Izvršit će se događaj putem klika na tipku C akcijom. C akcija predstavlja biblioteku različitih funkcija koja su zapisana u C programskom kodu. Prilikom odabira akcije pojavi se prozor sa C kodom gdje je omogućeno upisati i vlastitu funkciju neke radnje. Odabirom internih funkcija pronalazi se funkcija **ExitWinCC** koju se učita u prozor i time je definiranje radnje završeno.



Slika 2.22 Prikaz definiranja C akcije na klik

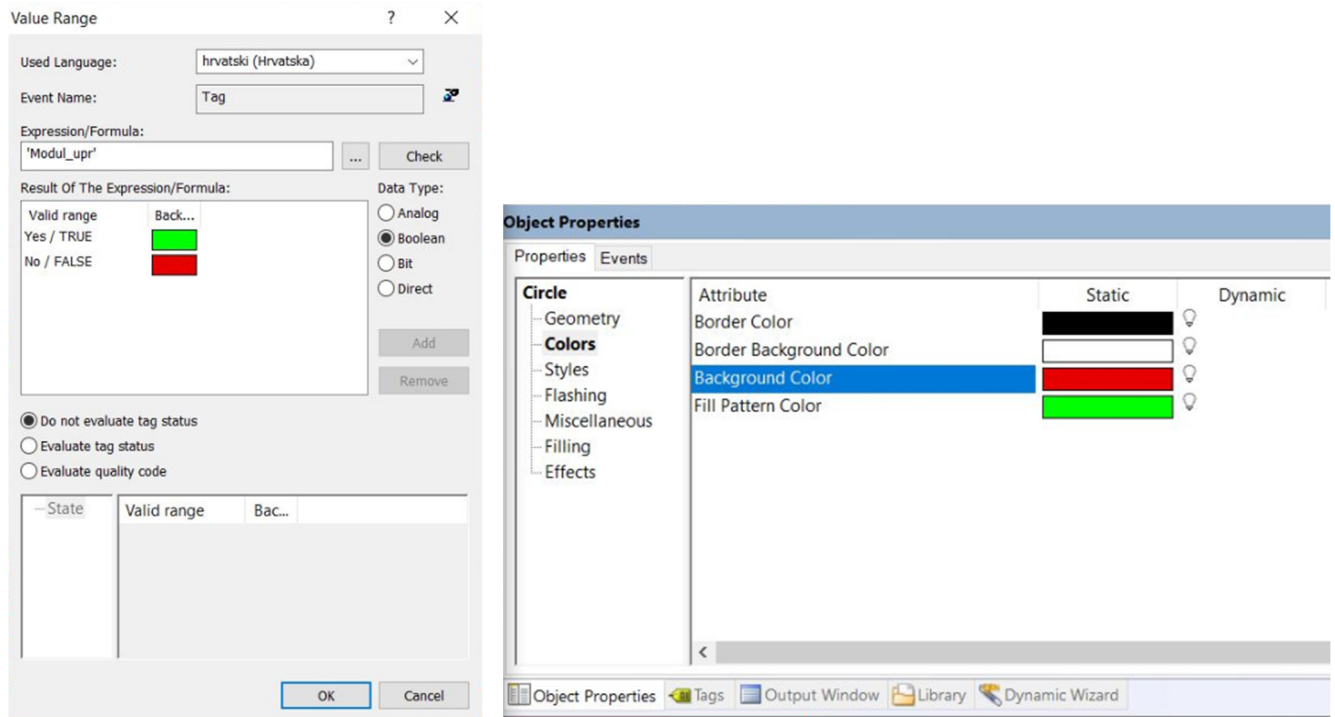
Desnim klikom na područje akcije „**Mouse Click**“ i odabir **C-action**.



Slika 2.23 Prikaz C programskog koda za izlaz iz programa na klik

Na slici „**Stanje upravljanja procesa**“ omogućen je pregled načina rada (Ručno putem modula, ručno putem WiFi-a ili automatski putem WiFi-a) kao i mogućnost izravnog djelovanja rada putem WinCC sustava na pojedini proces ili automatski režim rada.

Događaje, odnosno akcije prikazivanja definirane su putem dinamičkog prozora (Dynamic Dialog) ili direktnim povezivanjem stavke na tag (Kada je stanje 1 vrši se događaj na element).

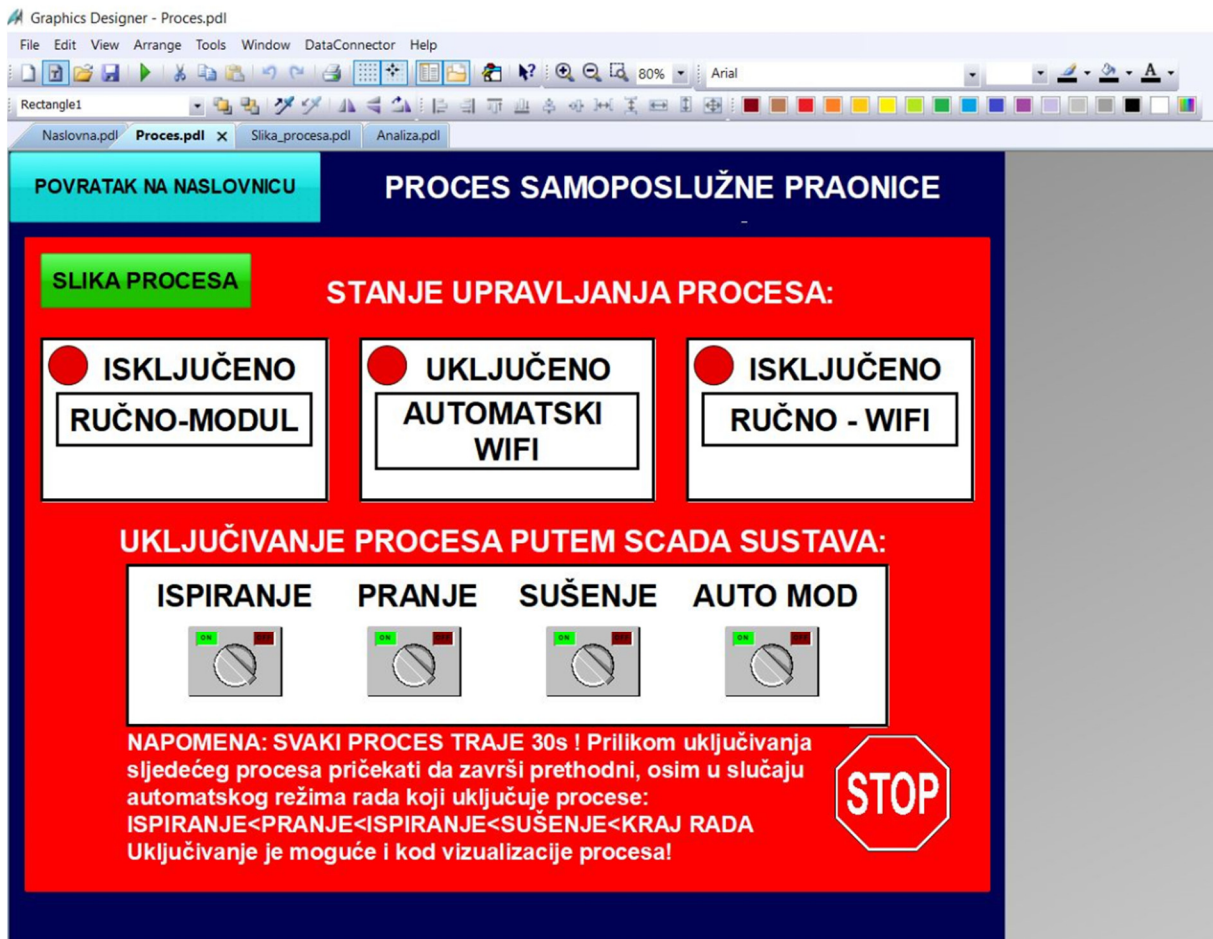


Slika 2.24 Prikaz određivanja događaja putem dinamičkog prozora

Na primjeru je rađeno prikazivanje lampice rada pojedinog procesa. U pravilu, ako je tag **Modul_upr** uključen izlazna boja elementa je zelena, ako je isključen izlazna boja je crvena.

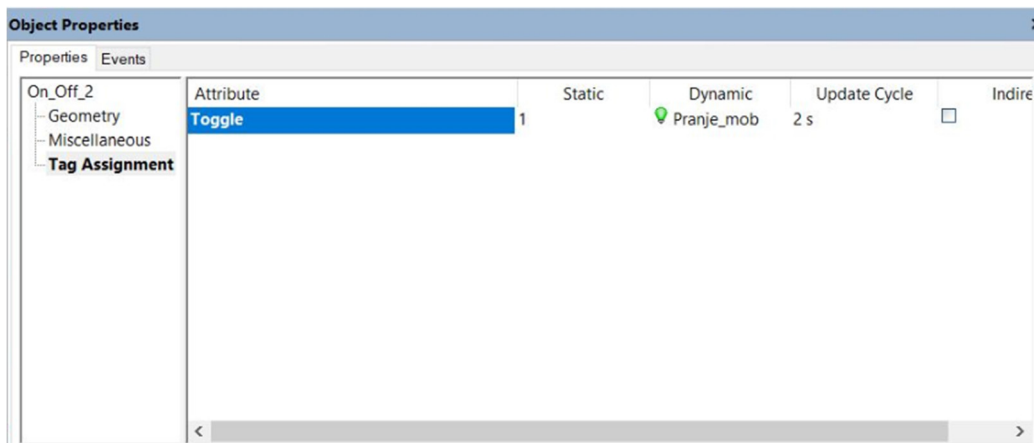
Tip podataka mora biti definiran u ovome slučaju „**Boolean**“

Kod korištenja boolean tip podataka omogućeno je predstavljanje dvije vrijednosti logike, u ovom slučaju ako daje **izlaz stanja 1** vrši se događaj **TRUE**, ako daje izlaz **stanja 0** vrši se događaj **FALSE**.



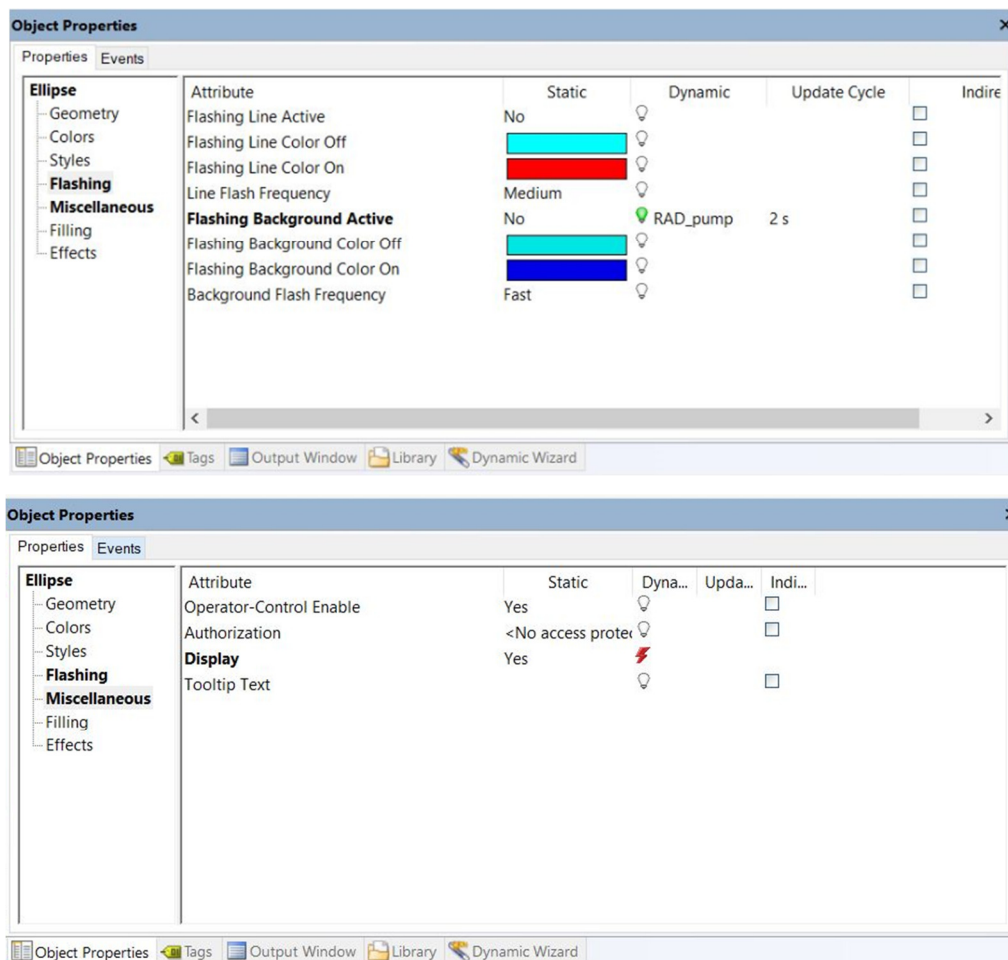
Slika 2.25 Prikaz slike stanja upravljanja procesa

Tipke za omogućen rad pojedinih procesa izravno iz WinCC sustava definirane su direktnim događajem tag-a. Dizajn tipke preuzet je iz knjižnice blokova pod „**Toogle buttons**“



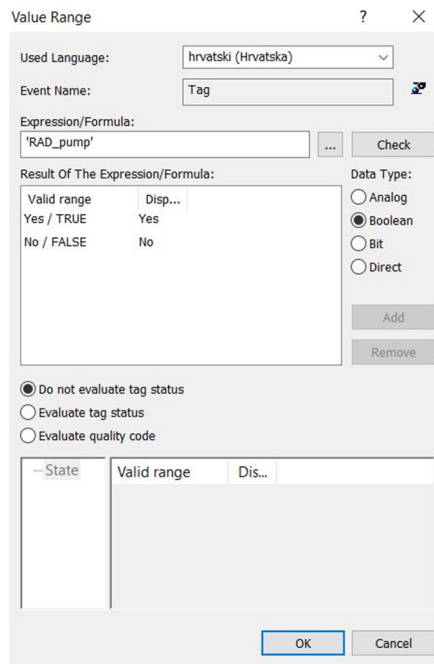
Slika 2.26 Prikaz definiranja rada tipke te prikaz tipke
Desnim klikom na **Dynamic** odabirom na **Tag** te odabir tag-a **Pranje_mob**.

Izrada same slike procesa napravljena je pomoću pojedinih objekata u knjižnici sa omogućenim događajima. Definirano je svako stanje procesa, točnije rada pojedinih komponenata te prilikom vizualiziranja moguće je djelovanje na proces. Stvorena je dodatna nova slika pod imenom „**Slika procesa**“

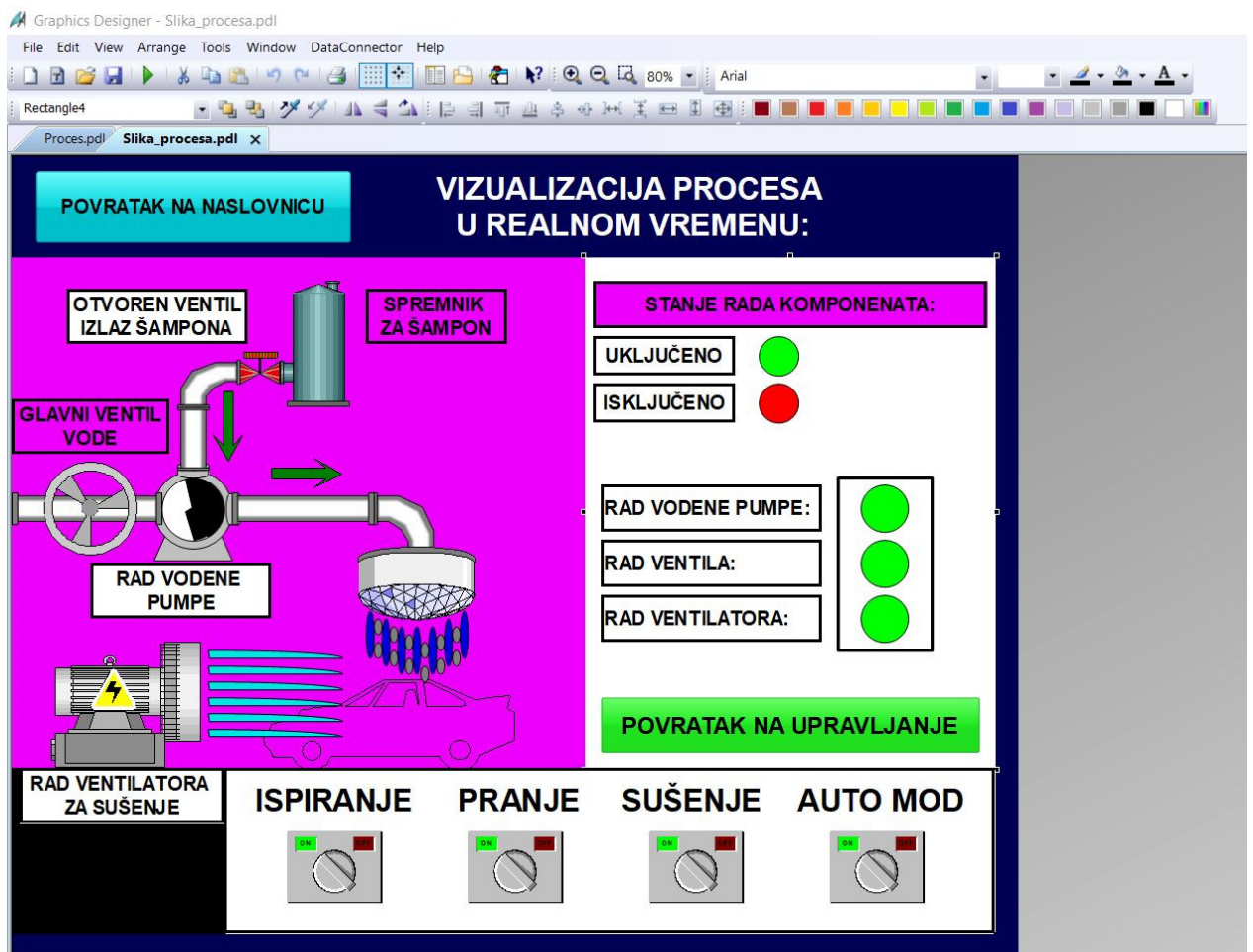


Slika 2.27 Prikaz definiranja događaja kod vizualizacije procesa

Kod fotografije 2.27 objekt ima mogućnost definiranja više događaja prema samom radu. Definirana je pomoću mogućnosti „**Flashing**“ promjenom boje pozadine prilikom uključenja tag-a RAD_pump (Tamno plava><Svjetlo plava). Boje se mijenjaju prema mogućnosti u brzom režimu. Drugi odabir događaja je u opciji „**Miscellaneous**“ kod odabira prikazivanja elemenata „Display“ putem dinamičkog prozora.

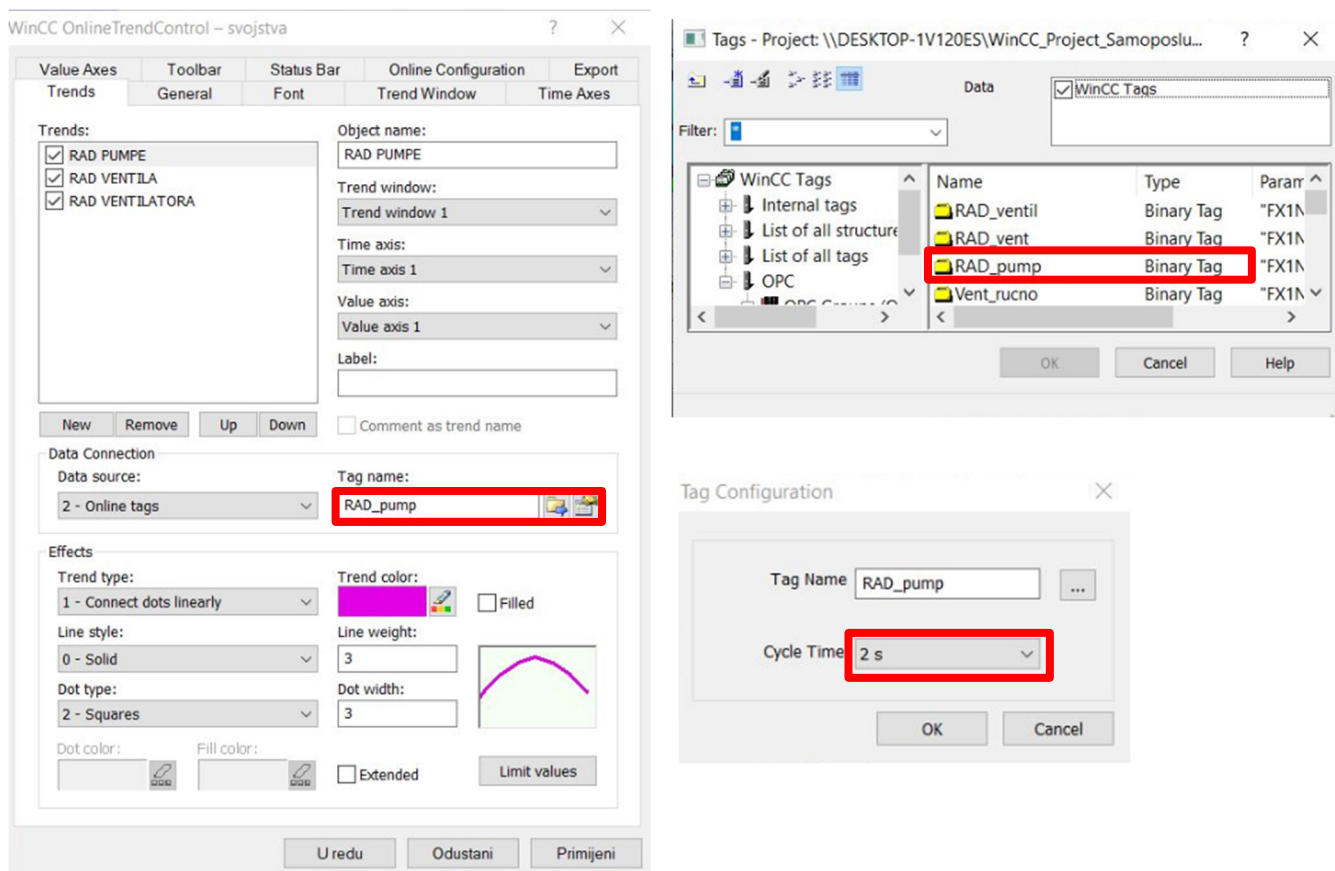


Slika 2.28 Prikaz definiranja događaja putem dinamičkog prozora



Slika 2.29 Prikaz slike vizualizacije procesa

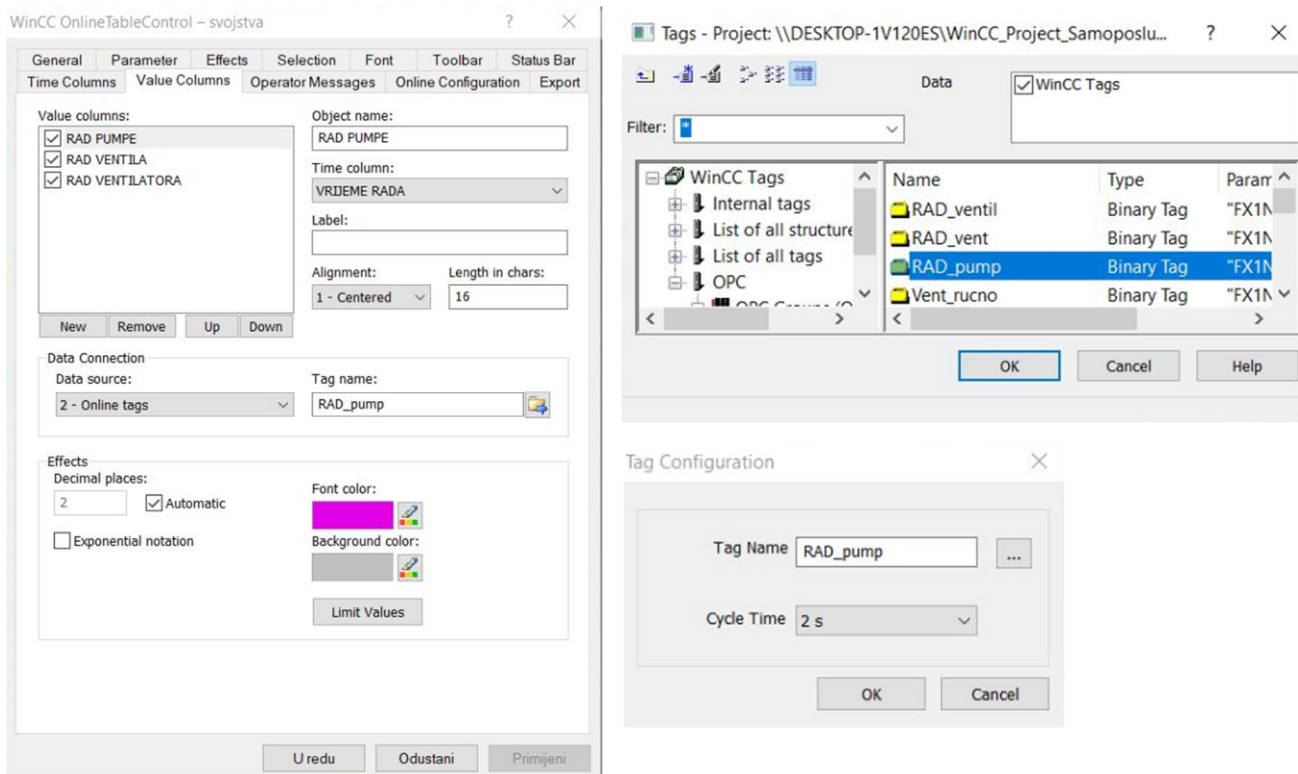
Na slici „Analiza upravljanja“ definirani su procesi kontrole **WinCC OnlineTrend Control** te **WinCC OnlineTable Control**.



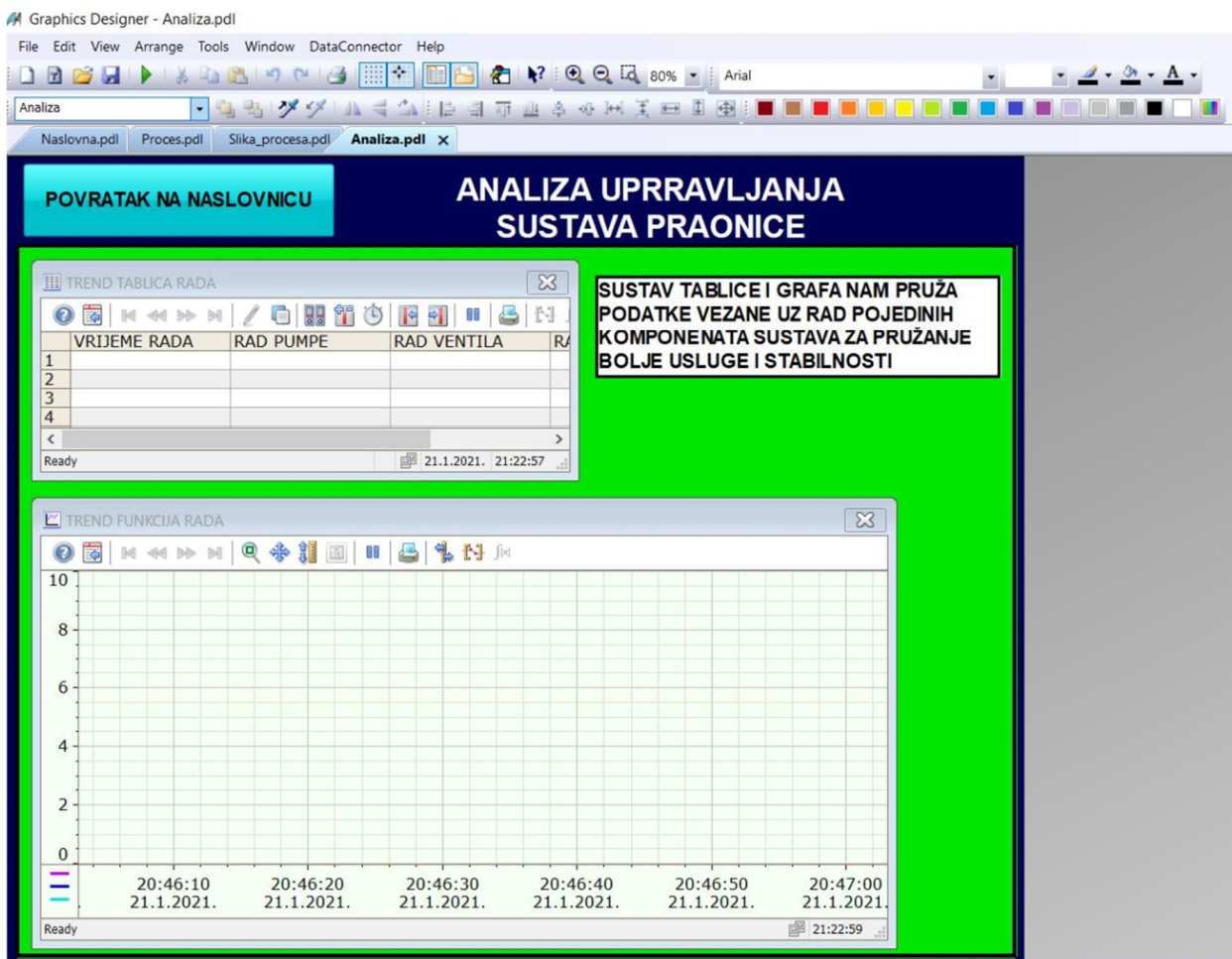
Slika 2.30 Prikaz definiranja parametra trenda grafičkog prikaza

Na slici mogu se uočiti definirane pojedine veličine putem tag-a te boje i naziva prikaza pojedinih grafova. Kod konfiguracije odabira tag-a može se odrediti ciklus ponavljanja očitavanja koji je određen očitanjem svakih 2s .

Kod uređivanja tabličnog prikaza situacija određivanja parametra je identična.



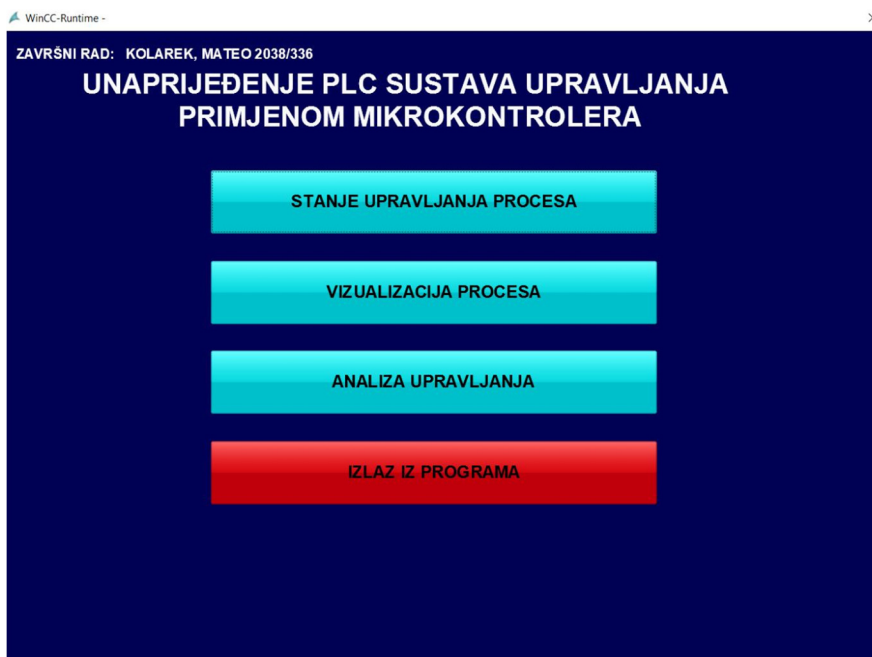
Slika 2.31 Prikaz definiranja parametra trenda tabličnog prikaza



Slika 2.32 Prikaz slike analize upravljanja

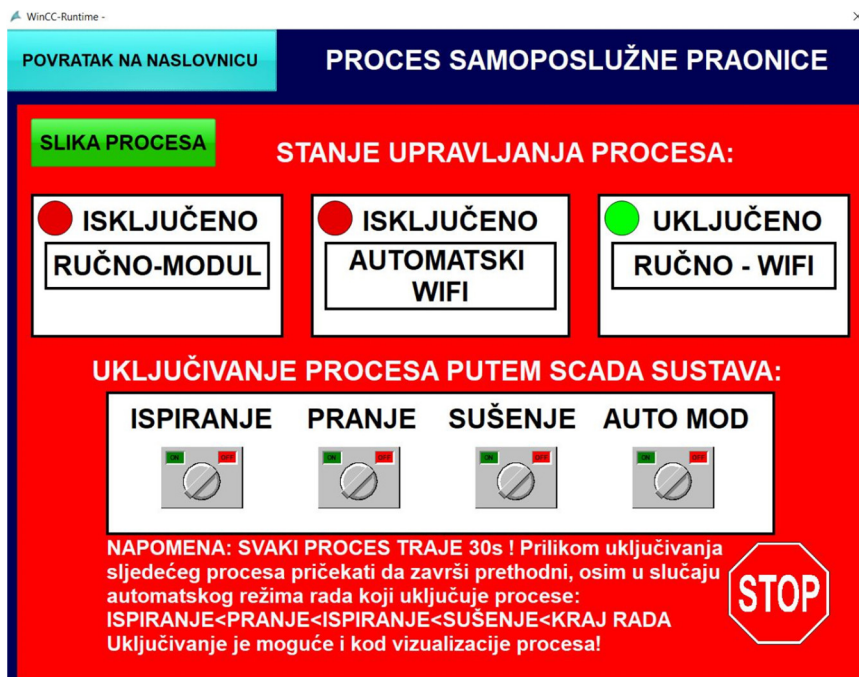
2.8. WinCC Runtime simulacija

Pokretanjem WinCC Runtime-a pokreće se simulacija samog procesa. Na naslovnoj slici moguće je putem klika od ponuđenih opcija otvoriti pojedinu sliku.



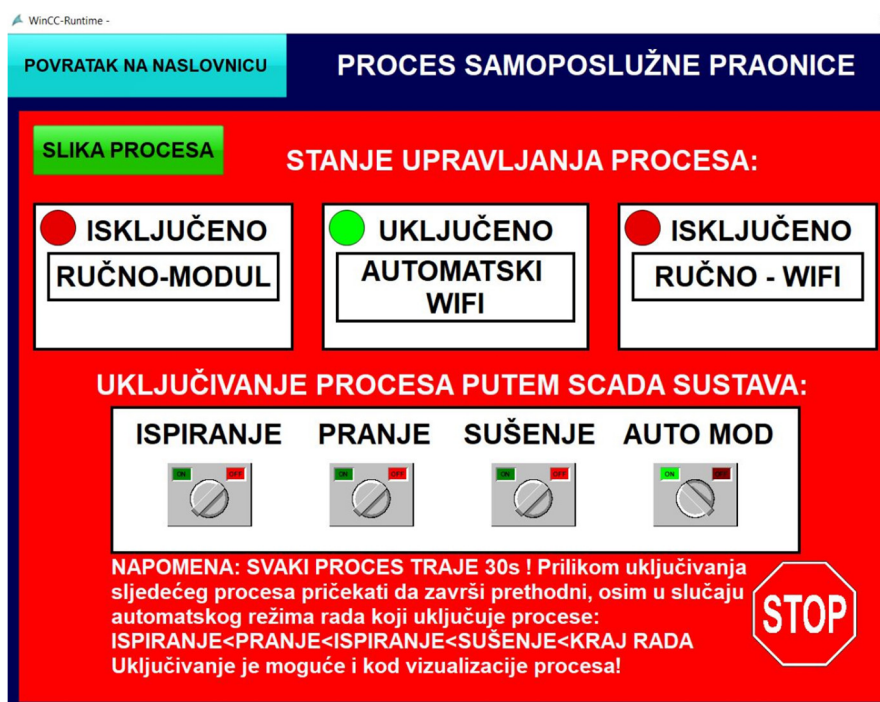
Slika 2.33 Prikaz naslovne slike prilikom simulacije

Odabirom „STANJE UPRAVLJANJA PROCESA“ otvara se slika gdje se može vidjeti trenutno stanje procesa, da li je u radu te kojim od ponuđenih načina upravljanja (ručno – putem modula, automatski ili ručno putem WiFi-a). Otvara se mogućnost pokretanja procesa putem WinCC-a odabirom na jedan od ponuđenih opcija uz napomenu.



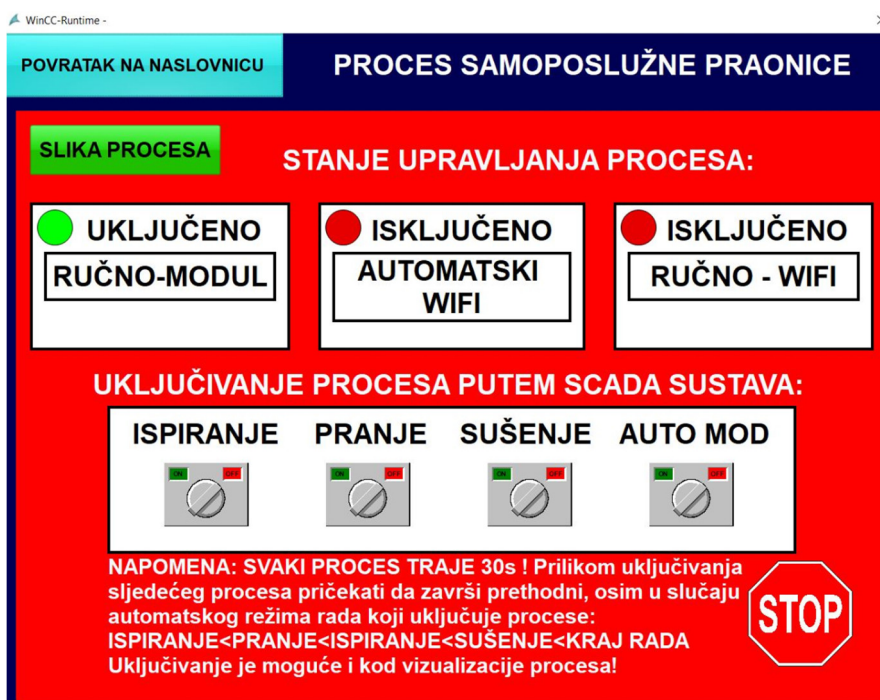
Slika 2.34 Prikaz stanja upravljanja prilikom simulacije

Uključenjem opcije „AUTO MOD“ uključuje se automatski režim rada putem mikrokontrolera koji sveukupno traje 120s.



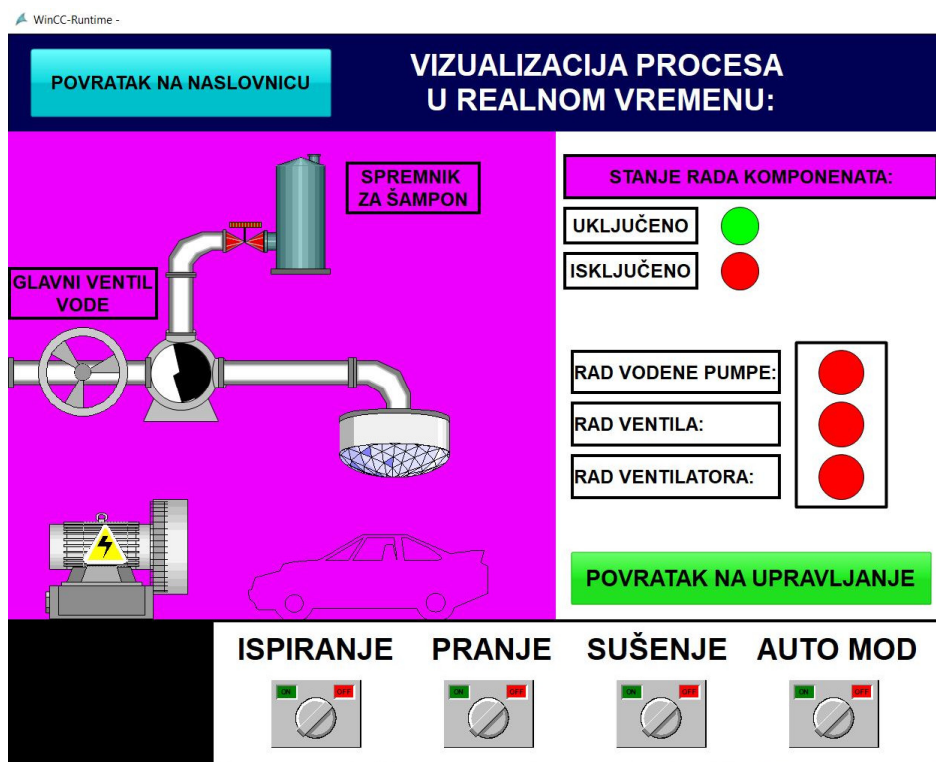
Slika 2.35 Prikaz stanja upravljanja prilikom simulacije automatskom rada

Uključenjem ručne sklopke X0.1 na modulu prikazuje se stanje uključenosti tog modula time svi procesi završavaju koji su trajali u tom trenutku t(s).



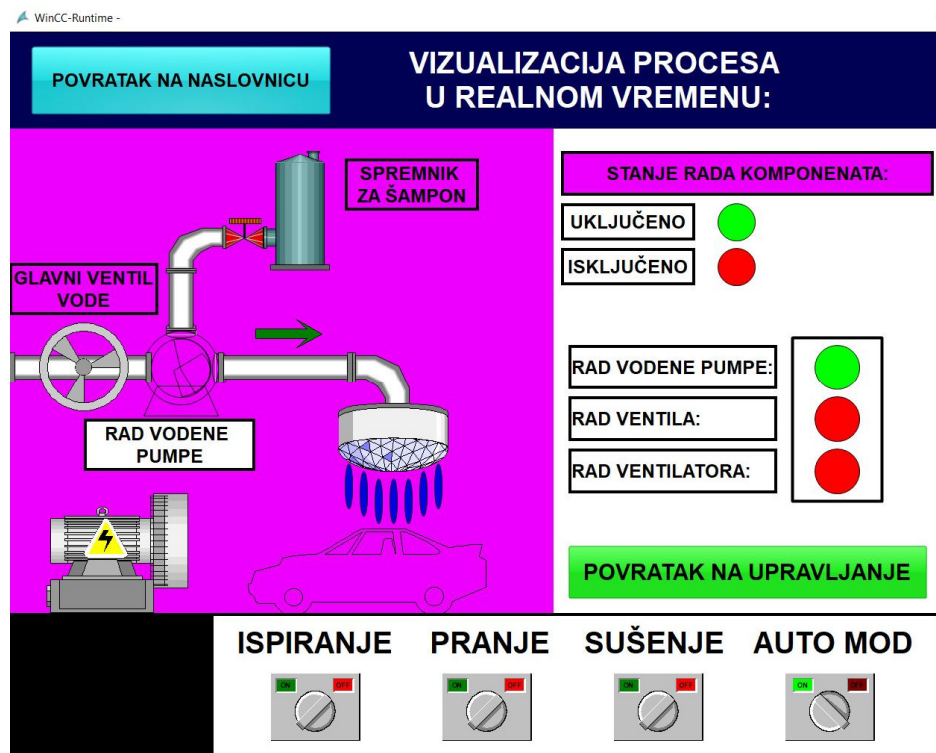
Slika 2.36 Prikaz stanja upravljanja prilikom simulacije rada ručnog modula

Odabirom na naslovnoj strani „VIZUALIZACIJA PROCESA“ otvara se slika same vizualizacije u kojoj je moguć pregled rada te odabir određenih procesa kao u prethodnoj slici.



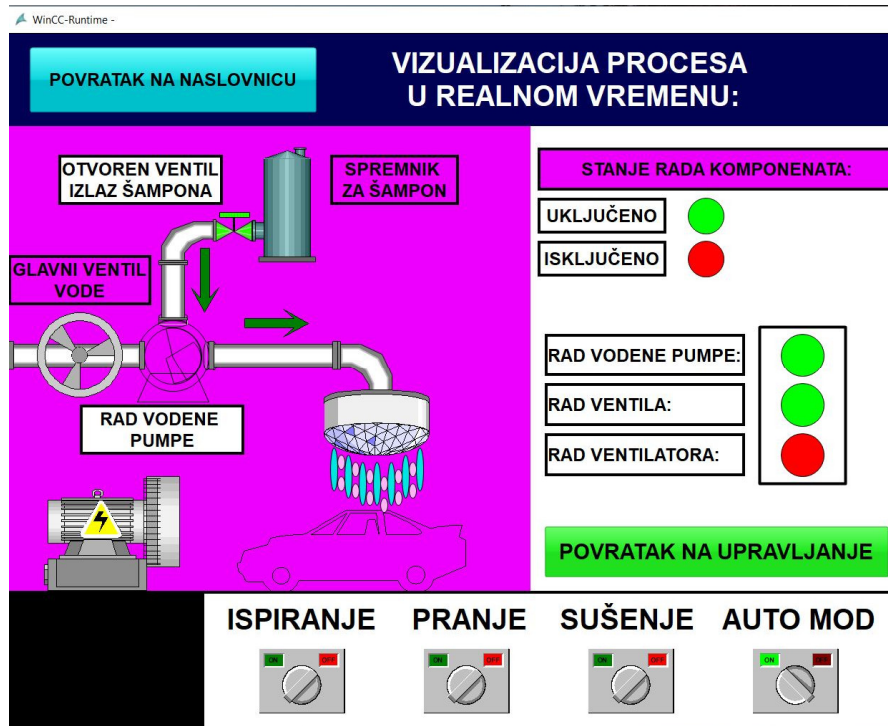
Slika 2.37 Prikaz slike procesa prilikom mirovanja u simulaciji

Uključenjem „AUTO MOD“ tipke pali se automatski režim rada i time započinje proces ispiranja, odnosno rad vodene pumpe.



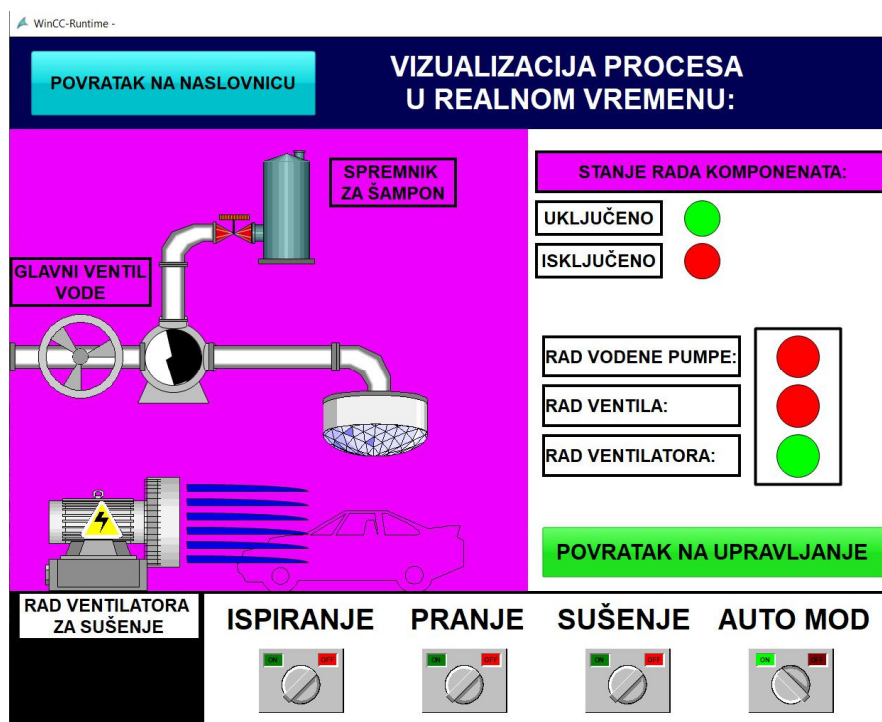
Slika 2.38 Prikaz slike procesa prilikom simulacije rada pumpe procesa ispiranja

Nakon završetka procesa ispiranja u automatskom režimu rada dolazi do druge faze, a to je proces pranja. U tom trenutku vizualizacija će pokazati rad vodene pumpe i otvorenog ventila koji ispušta šampon. Nakon procesa pranja dolazi do ponovnog procesa ispiranja.



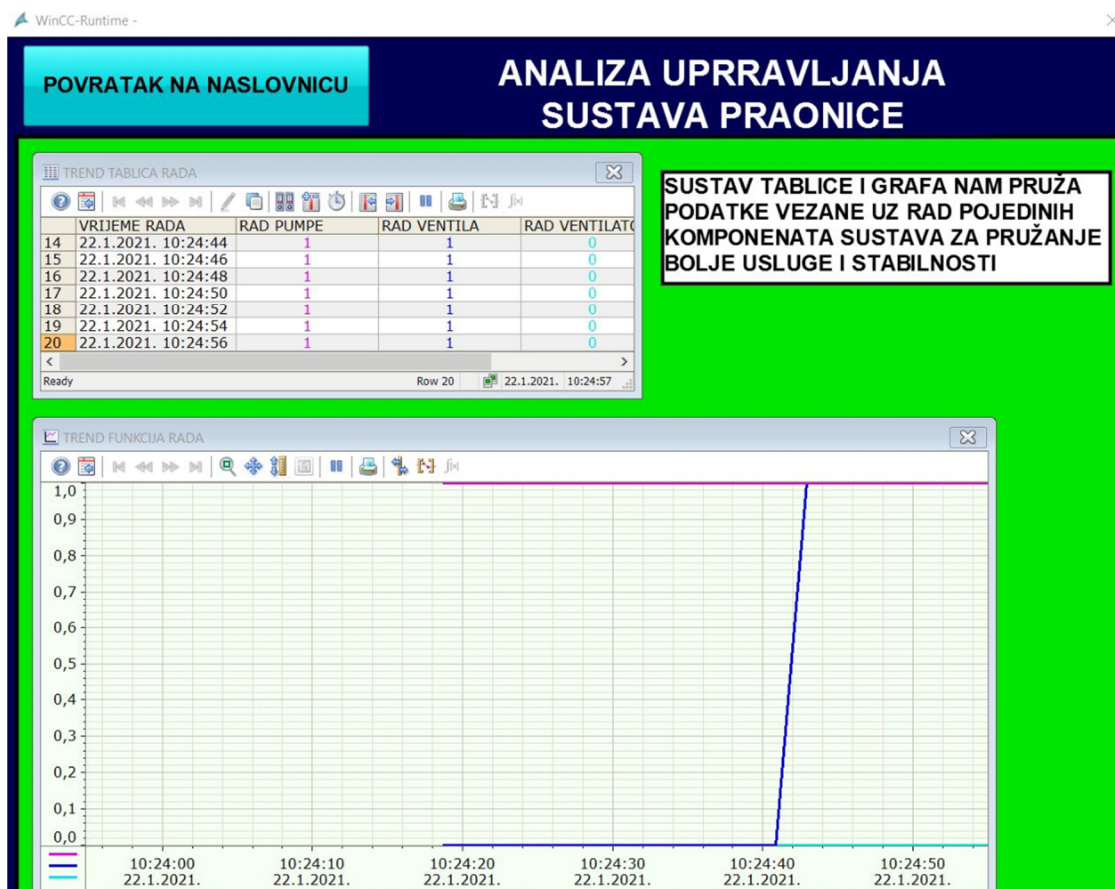
Slika 2.39 Prikaz slike procesa pranja

Za kraj u automatskom režimu rada dolazi do procesa sušenja, odnosno rada ventilatora i time automatski režim rada završava.



Slika 2.40 Prikaz slike procesa sušenja

Odabirom „ANALIZE UPRAVLJANJA SUSTAVA PRAONICE“ mogući je grafički i tablični uvid nad radom pojedinih komponenata procesa, što omogućuje znatno lakši uvid te otkrivanje greške, odnosno kvara.



Slika 2.41 Prikaz analize upravljanja prilikom simulacije iz procesa ispiranja u proces pranja

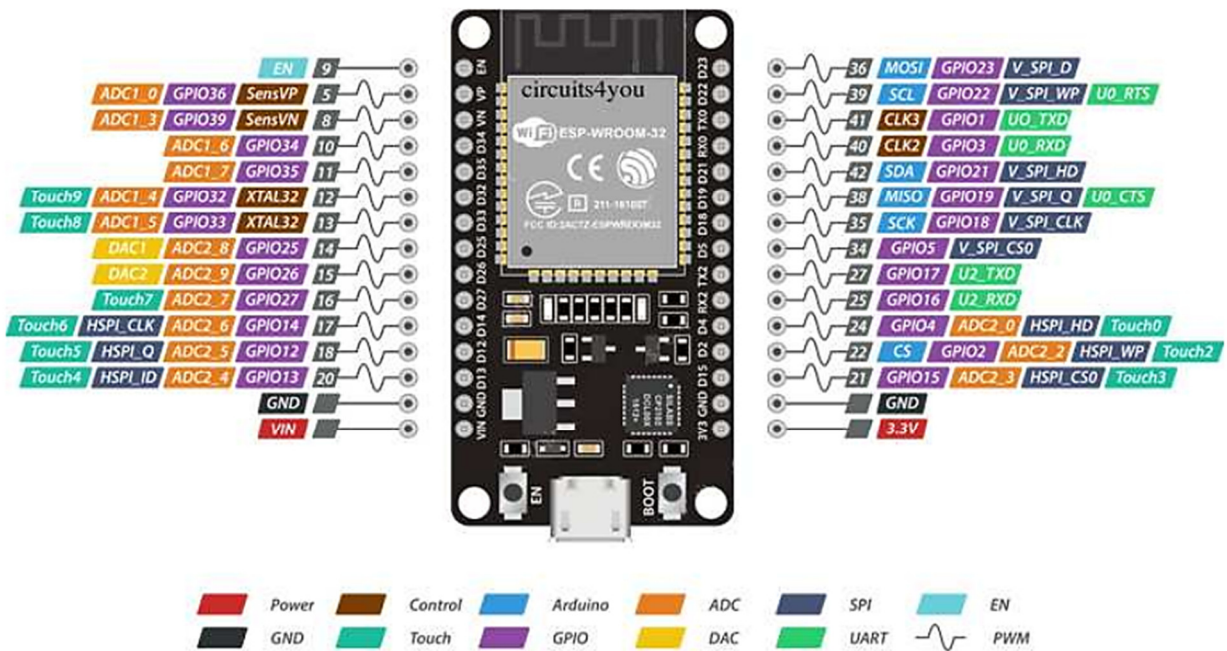
2.9. Karakteristike SCADA sustava

Karakteristike SCADA sustava su što je vrlo jednostavno moguće generirati neku osnovnu vizualizaciju, ali mora se imati na umu da što detaljnije izrađena vizualizacija sa definiranim alarmnim obavijestima omogućuje brže i efikasnije djelovanje za razliku od „grube“ vizualizacije sa osnovnom izvedbom. Na raspolaganju uvijek treba biti preventivno održavanje za otkrivanje i otklanjanje mogućih grešaka, kao i mogućnost promjene prilagođenih komponenti, nadogradnje, testiranja ili poboljšanja. Za svaku funkciju SCADA sustava treba biti definirano vrijeme odziva, zbog djelovanja u skladu sa načinom rada sustava i upravljanja. Prilikom izrade SCADA sustava važan je način komunikacije i slanje podataka. Većina računala ima mogućnost mrežne komunikacije. Prilikom projektiranja takvih sustava važno je obratiti pažnju na sigurnost pristupa te slanja podataka. Važno je definirati sigurnosni pristup za dohvat podataka, zbog toga što neki podaci mogu biti raspoloživi širokom krugu korisnika, dok neki podaci moraju imati ograničeni pristup zbog mogućih opasnosti.

3. Upravljanje sustava pomoću ESP-WROOM-32

3.1. ESP-WROOM-32

ESP32 je veoma „popularan“ mikrokontroler s integriranim Wi-Fi-em i dual-mode Bluetoothom. Serija ESP32 koristi mikroprocesor Tensilica Xtensa LX6 u dual-core i single-core varijacijama te uključuje i ugrađene antenske prekidače, pojačalo snage, različite filtere te module za upravljanje. ESP32 je kreirao i razvio Espressif Systems, kineska tvrtka sa sjedištem u Šangaju, a proizvodi ga TSMC. Prethodnik ovog mikrokontrolera je ESP8266.



Slika 3.1 Prikaz pinova ESP32 mikrokontrolera [12]



Slika 3.2 Prikaz ESP32 mikrokontrolera [12]

3.2. Upravljanje pomoću ESP-WROOM-32

U radu je bilo potrebno implementirati mogućnost upravljanja PLC uređaja pomoću ESP32 mikrokontrolera putem WiFi-a preko pametnog telefona. Korištenjem takve tehnologije omogućuje se fleksibilno upravljanje i nadzor nad nekim procesom. Najčešća takva upotreba je efikasna kod poslužitelja i klijenta. Klijent stvara svoju odluku putem privatnog uređaja kojim preko WiFi-a, Bluetooth-a ili neke druge vrste komunikacije šalje povratne informacije odnosno podatke koje se direktno implementiraju na nekakav postojeći automatizirani sustav kao u primjeru samoposlužne praonice.

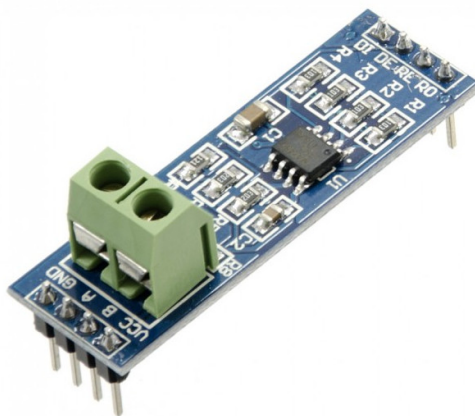
Izradit će se HTML obrazac u Arduino okruženju preko kojeg će biti moguće, putem ESP32 mikrokontrolera utjecati na tijek određenog procesa u PLC uređaju.

3.3. Povezivanje ESP-WROOM-32 sa FX1N-32MR

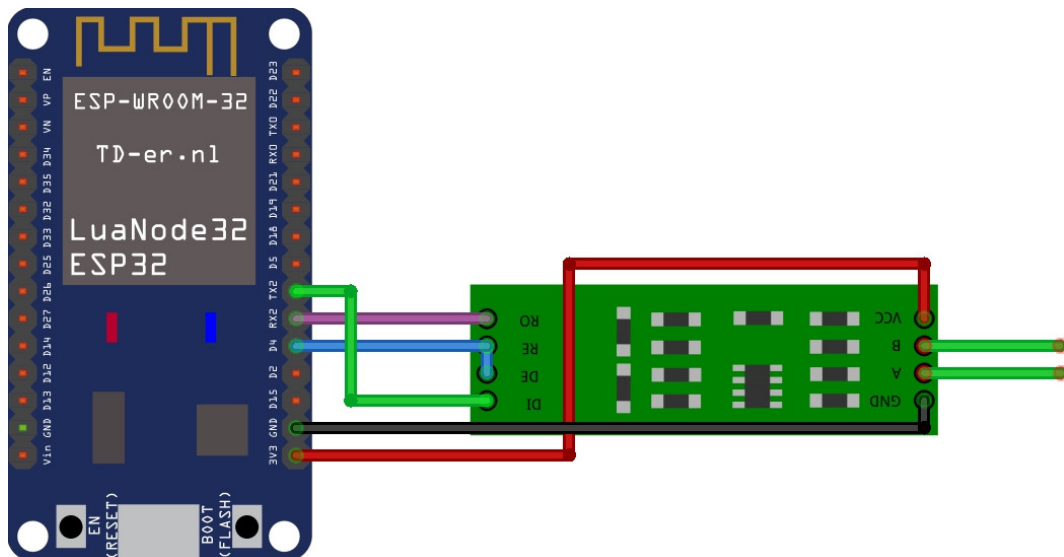
ESP32 je moguće povezati na više načina sa PLC uređajem, u ovome slučaju opisan će se dvije vrste mogućih povezivanja sa FX1N-32MR PLC uređajem.

1. Putem RS-485 standard digitalne komunikacije

Za povezivanje putem RS-485 komunikacije može se koristiti međusklop TTL-RS485 pretvarač preko kojeg se može serijski povezati ESP32 mikrokontroler s PLC uređajem.



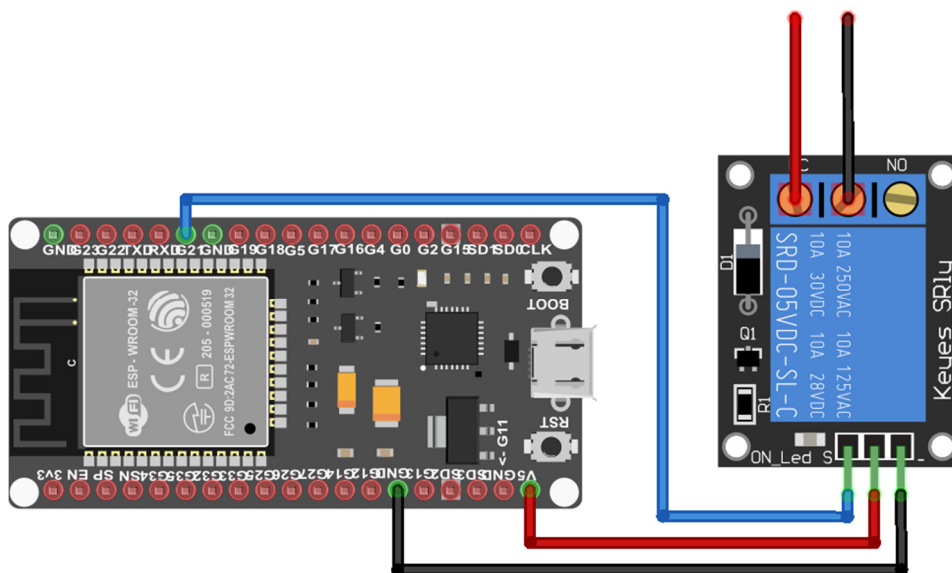
Slika 3.3 Prikaz TTL-RS485 pretvarača [14]



Slika 3.4 Prikaz povezivanja ESP32 mikrokontrolera sa TTL-RS485 [16]

2. Direktna povezanost putem releja

Direktna povezanost putem releja ne predstavlja komunikaciju između mikrokontrolera i PLC-a već predstavlja direktno djelovanje na ulazni dio PLC-a, što se može vizualizirati te definirati pomoću programa. Takav pristup nije posve ekonomičan zbog povećanja relejnih upravljačkih jedinica.



Slika 3.5 Prikaz povezivanja ESP32 mikrokontrolera sa Arduino relejem (3.3-5V) [17]

3.4. Programsko rješenje

Putem ESP32 mikrokontrolera implementirat će se mogućnost odabira procesa samoposlužne praonice putem mobilnog korisničkog sučelja. Za izradu programskog rješenja koristit će se platforma „Arduino“.

```
/**
 * Program baziran na završnom projektu putem ESP32 kontrolera za komunikaciju slanja
 * na PLC
 * ZAVRSNI RAD: MATEO KOLAREK 2038/336
 **/

// Ucitavanje biblioteke Wifi za povezivanje Wifi-em.
#include <WiFi.h>

// Podaci za pristup WIFI mrezi
const char* ssid = "MATRIX";
const char* password = "#####";

// Postavljanje porta 80
WiFiServer server(80);
```

Programski blok 1 Prikaz programa za definiranje biblioteke, WiFi pristupa te postavljane porta

<WiFi.h> je biblioteka koja omogućuje povezivanje nekog arduino mikrokontrolera s internetom. Također može poslužiti kao poslužitelj koji prihvaća dolaznu vezu ili kao klijent koji uspostavlja odlaznu vezu. Biblioteka podržava WEP i WPA2 šifriranje.

Arduino komunicira s WiFi-om pomoću SPI sabirnice. WiFi biblioteka je slična Ethernet biblioteci, mnogo njihovih funkcija su identične.

WiFiServer server(80); time se definira poslužitelj koji reagira na klijente, odnosno u ovome slučaju web preglednik koji komunicira s web poslužiteljem.

```
// Definiranje varijable za pohranu HTTP request-a
String header;

// Definiranje varijabli za trenutnu pohranu stanja pinova
String output33State = "off";
String output25State = "off";
String output26State = "off";
String output27State = "off";

// Definiranje varijabli za GPIO pinove
const int output33 = 33;
const int output25 = 25;
const int output26 = 26;
const int output27 = 27;
```

```

// Trenutno vrijeme
unsigned long currentTime = millis();
// Prethodno vrijeme
unsigned long previousTime = 0;
// Definirano u milisekundama (primjer: 2000ms = 2s)
const long timeoutTime = 2000;

```

Programski blok 2 Prikaz programa za definiranje izlaznih pinova

Koristit će se četiri izlazna pinova (*GPIO33, GPIO25, GPIO26 i GPIO27*) koja će djelovati na ulaze PLC-a. Početno stanje pinova postavljeno je u digitalnu nulu.

```

void setup() {
  Serial.begin(115200);
  // Inicijalizacija izlaznih varijabli
  pinMode(output33, OUTPUT);
  pinMode(output25, OUTPUT);
  pinMode(output26, OUTPUT);
  pinMode(output27, OUTPUT);
  // Postavljanje vanjskih pinova na nisko stanje 0
  digitalWrite(output33, LOW);
  digitalWrite(output25, LOW);
  digitalWrite(output26, LOW);
  digitalWrite(output27, LOW);

  // Povezivanje na WIFI mrežu sa podacima
  Serial.print("Povezivanje na ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Ispis lokalne IP adrese preko koje se povezujemo za pristup
  Serial.println("");
  Serial.println("WiFi povezan.");
  Serial.println("IP adresa: ");
  Serial.println(WiFi.localIP());
  server.begin();
}

```

Programski blok 3 Prikaz programa za povezivanje na mrežu te ispis podataka.

Naredbom `Serial.begin(115200);` postavlja se brzina prijenosa podataka u bitovima po sekundi (baud) za serijski prijenos podataka. Za komunikaciju sa serijskim monitorom obavezno se koristi jedna od ponuđenih brzina prijenosa u izborniku, ali nije neophodno.

```

void loop(){
  WiFiClient client = server.available(); // Priprema za klijenta

  if (client) { // U slucaju povezivanja novog klijenta
    currentTime = millis();
    previousTime = currentTime;
    Serial.println("New Client."); // Ispis poruke u serijskom monitoru
    String currentLine = ""; // Izrada stringa za nadolazece podatke
    klijenta
    while (client.connected() && currentTime - previousTime <= timeoutTime) { //
    petlja while sve do povezivanja s klijentom
      currentTime = millis();
      if (client.available()) { // Ako postoje podaci za citanje kod
    klijenta
        char c = client.read(); // Citaj podatke
        Serial.write(c); // Ispis podataka na serijskom monitoru
        header += c;
        if (c == '\n') { // Ako je podatak drugi karakter
          // Kraj HTTP zahtjeva, time se salje odgovor:
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Povezivanje: ZAVRSENO");
            client.println();

```

Programski blok 4 Prikaz programa za slanje i primanje podataka između klijenta i poslužitelja

WiFiClient client = server.available(); Povezivanjem će dobiti pristup samo klijent koji je povezan sa istim poslužiteljem te dobiva dostupne podatke za čitanje, odnosno obradu.

```

// Uključujemo GPIO pinove
if (header.indexOf("GET /33/on") >= 0) {
  Serial.println("GPIO 33 on");
  output33State = "UKLJUCENO";
  digitalWrite(output33, HIGH);
} else if (header.indexOf("GET /33/off") >= 0) {
  Serial.println("GPIO 33 off");
  output33State = "ISKLJUCENO";
  digitalWrite(output33, LOW);
} else if (header.indexOf("GET /25/on") >= 0) {
  Serial.println("GPIO 25 on");
  output25State = "UKLJUCENO";
  digitalWrite(output25, HIGH);
} else if (header.indexOf("GET /25/off") >= 0) {
  Serial.println("GPIO 25 off");
  output25State = "ISKLJUCENO";
  digitalWrite(output25, LOW);
} else if (header.indexOf("GET /26/on") >= 0) {
  Serial.println("GPIO 26 on");
  output26State = "UKLJUCENO";
  digitalWrite(output26, HIGH);
} else if (header.indexOf("GET /26/off") >= 0) {
  Serial.println("GPIO 26 off");
  output26State = "ISKLJUCENO";
  digitalWrite(output26, LOW);

```

```

} else if (header.indexOf("GET /27/on") >= 0) {
  Serial.println("GPIO 27 on");
  output27State = "UKLJUCENO";
  digitalWrite(output27, HIGH);
} else if (header.indexOf("GET /27/off") >= 0) {
  Serial.println("GPIO 27 off");
  output27State = "ISKLJUCENO";
  digitalWrite(output27, LOW);
}

```

Programski blok 5 Prikaz programa za ispis stanja pinova prilikom promjene stanja

.indexOf Služi za lociranje niza unutar drugog niza. Prema postavkama pretraživanje se vrši od početka niza, ali može i započeti od određenog indeksa, što olakšava pretragu.

```

// Prikaz HTML stranice
client.println("<!DOCTYPE html><html>");
client.println("<head><meta name=\"viewport\" content=\"width=device-
width, initial-scale=1\">");
client.println("<link rel=\"icon\" href=\"data:,\>");
// CSS dizajn teksta, tipkala te pozadine.
client.println("<style>html { background-color: darkblue; opacity: 0.9;
font-family: Monospace; display: inline-block; margin: 0px auto; text-align: center;
color: white;});");
client.println(".button { display: inline-block; padding: 15px 25px;
font-size: 24px; cursor: pointer; text-align: center; text-decoration: none;
outline: none; color: #fff; background-color: #12e91b; border: none; border-radius:
15px; box-shadow: 0 9px #999;});");
client.println("text-decoration: none; font-size: 30px; margin: 2px;
cursor: pointer;});");
client.println(".button2 {background-color: #ea2622; box-shadow: 0 5px
#666; transform: translateY(4px);}</style></head>");

```

Programski blok 6 Prikaz programa za ispis oblika HTML stranice

Da bi se olakšao pristup programiranju, smješten je oblik uređenja HTML (*eng. Hyper Text Markup Language*) obrasca unutar Arduino okruženja.

client.println("<!DOCTYPE html><html>"); Omogućuje se početak rada u HTML obrascu kao i samostalno napravljeni HTML kod na donjoj slici u programu Visual Studio Code.

Naredbom html:5 (TAB) automatski je moguće dobiti ovu strukturu za daljnji rad HTML-a.

```

<!DOCTYPE html>
<html lang="hr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dokument</title>
</head>
<body>

</body>
</html>

```

Programski blok 7 Prikaz osnovne strukture HTML obrasca

Dizajn mobilnog korisničkog sučelja definira se osnovnim CSS (eng. *Cascading Style Sheets*)

parametrima:

```
background-color: darkblue;
opacity: 0.9;
font-family: Monospace;
display: inline-block;
margin: 0px auto;
text-align: center;
color: white;
```

```
// Naslon WEB STRANICE
client.println("<body><h1><b>UNAPRJEJENJE PLC SUSTAVA UPRAVLJANJA
PRIMJENOM MIKROKONTROLERA<b/></h1>");
```

Programski blok 8 Prikaz definiranog naslova web stranice

```
// Prikaz trenutnog stanja, START/STOP tipka za pin GPIO 33
client.println("<p>AUTOMATSKI RAD - " + output33State + "</p>");
// Uvjeti rada if else if funkcijom kod ukljucivanja i iskljucivanja
AUTOMATSKOG RADA
if ((output33State=="ISKLJUCENO") && (output26State=="ISKLJUCENO") &&
(output27State=="ISKLJUCENO") && (output25State=="ISKLJUCENO")) {
client.println("<p><a href=\"/33/on\"><button class=\"button\">START
AUTOMATSKI</button></a></p>"); }
else if (output33State=="UKLJUCENO")
{client.println("<p><a href=\"/33/off\"><button class=\"button
button2\">START RUCNO</button></a></p>"); }
else {client.println("<p><a href=\"/33/off\"><button class=\"button
button3\">NEDOSTUPNO</button></a></p>"); }

// Prikaz trenutnog stanja, START/STOP tipka za pin GPIO 25
client.println("<p>STANJE RADA ISPIRANJA - " + output25State + "</p>");
// Uvjeti rada if else if funkcijom kod ukljucivanja i iskljucivanja
ISPIRANJA
if ((output33State=="ISKLJUCENO") && (output25State=="ISKLJUCENO") &&
(output26State=="ISKLJUCENO") && (output27State=="ISKLJUCENO")) {
client.println("<p><a href=\"/25/on\"><button class=\"button\">START
ISPIRANJE</button></a></p>"); }
else if (output25State=="UKLJUCENO")
{client.println("<p><a href=\"/25/off\"><button class=\"button
button2\">STOP ISPIRANJE</button></a></p>"); }
else {client.println("<p><a href=\"/25/off\"><button class=\"button
button3\">NEDOSTUPNO</button></a></p>"); }

// Prikaz trenutnog stanja, START/STOP tipka za pin GPIO 26
client.println("<p>STANJE RADA PRANJA - " + output26State + "</p>");
// Uvjeti rada if else if funkcijom kod ukljucivanja i iskljucivanja
PRANJA
if ((output33State=="ISKLJUCENO") && (output26State=="ISKLJUCENO") &&
(output25State=="ISKLJUCENO") && (output27State=="ISKLJUCENO")) {
```

```

        client.println("<p><a href=\"/26/on\"><button class=\"button\">START
PRANJE</button></a></p>"); }
    else if (output26State=="UKLJUCENO")
        {client.println("<p><a href=\"/26/off\"><button class=\"button
button2\">STOP PRANJE</button></a></p>"); }
    else {client.println("<p><a href=\"/26/off\"><button class=\"button
button3\">NEDOSTUPNO</button></a></p>"); }

// Prikaz trenutnog stanja, START/STOP tipka za pin GPIO 27
    client.println("<p>STANJE RADA SUSENJA - " + output27State + "</p>");
    // Uvjeti rada if else if funkcijom kod ukljucivanja i iskljucivanja
SUSENJA
    if ((output33State=="ISKLJUCENO") &&(output27State=="ISKLJUCENO") &&
(output25State=="ISKLJUCENO") && (output26State=="ISKLJUCENO")) {
        client.println("<p><a href=\"/27/on\"><button class=\"button\">START
SUSENJE</button></a></p>"); }
    else if (output27State=="UKLJUCENO")
        {client.println("<p><a href=\"/27/off\"><button class=\"button
button2\">STOP SUSENJE</button></a></p>"); }
    else {client.println("<p><a href=\"/27/off\"><button class=\"button
button3\">NEDOSTUPNO</button></a></p>"); }

```

Programski blok 9 Prikaz programa za definiranje odabira rada procesa korisničkog sučelja

Svaki dio u korisničkom sučelju je integriran kao tipka „Button“ kojim je moguće upravljati na dva načina („Button2“). Uključivanjem putem odgovora na funkciju za prvi integritet automatskog ili ručnog odabira :

```

if ((output33State=="ISKLJUCENO") &&(output26State=="ISKLJUCENO") &&
(output25State=="ISKLJUCENO") && (output27State=="ISKLJUCENO"))

```

Koja govori:

Ako je stanje pina **GPIO33** na izlazu **stanje 0**

I (operator)

Ako je stanje pina **GPIO26** na izlazu **stanje 0**

I (operator)

Ako je stanje pina **GPIO25** na izlazu **stanje 0**

I (operator)

Ako je stanje pina **GPIO27** na izlazu **stanje 0**

Omogući „START AUTOMATSKI“

```

client.println("<p><a href=\"/33/on\"><button class=\"button\">START
AUTOMATSKI</button></a></p>");

```

```

else if (output33State=="UKLJUCENO")

```

Ako je stanje pina **GPIO33** na izlazu **stanje 1**

Omogući „START RUCNO“

```
client.println("<p><a href=\"/33/off\"><button class=\"button  
button2\">START RUCNO</button></a></p>");
```

U slučaju ako je neki drugi uvjet zadovoljen, u ovome slučaju ova funkcija ne bi trebala biti u mogućnosti za korištenje: **GPIO33** na izlazu **stanje 0**, nemogući „**NEDOSTUPNO**“

```
else {client.println("<p><a href=\"/33/off\"><button class=\"button  
button3\">NEDOSTUPNO</button></a></p>");}
```

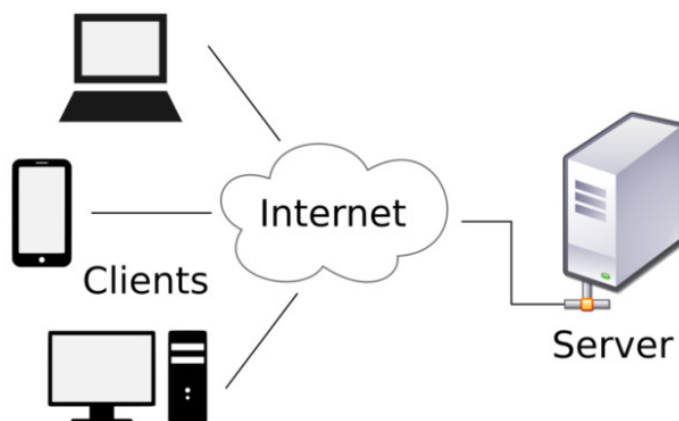
```
// HTTP odgovor završava  
    client.println();  
    // Prekid  
    break;  
} else {  
    currentLine = "";  
}  
} else if (c != '\r') {  
    currentLine += c;  
}  
}  
}  
  
header = "";  
// Zatvaranje komunikacije  
client.stop();  
Serial.println("Odjava Klijenta.");  
Serial.println("");  
}  
}
```

Programski blok 10 Prikaz programa za definiranje HTTP odgovara – Kraj programa

3.5. Komunikacija ESP32 i klijentsko-poslužiteljska veza

U današnje vrijeme svakodnevno se susreće putem internet preglednika izraz „klijent“ i „poslužitelj“ kao odnos funkcioniranja između našeg računala ili nekog uređaja koji ima pristup internetu prema nekom serveru, odnosno pristupnoj točki informacija.

Server je računalo koje pruža podatke drugim računalima koja putem interneta pristupaju iz lokalne mreže ili širokopojasne mreže. Postoje mnogo vrsti poslužitelja, svako prema svojim zadanim dužnostima. S druge strane postoji klijent, koji preuzima podatke s poslužitelja. Npr. kada se provjerava e-pošta na računalu. U tome trenutku računalo postaje klijent koji pristupa podacima s poštanskog poslužitelja.



Slika 3.6 Prikaz odnosa klijent-poslužitelj [18]

Klijentsko-poslužiteljski odnos u ovom radu je automatski sustav baziran na lokalnoj mreži. Putem WiFi pristupne točke ESP32 mikrokontroler prima tj. šalje podatke računalu ili pametnom telefonu.


```
Done uploading.
Sketch uses 643486 bytes (49%) of program storage space. Maximum is 1310720 bytes.
Global variables use 38840 bytes (11%) of dynamic memory, leaving 288840 bytes for local variables. Maximum is 327680 bytes.
esptool.py v2.6
Serial port COM8
Connecting.....
Chip is ESP32D0WD06 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, Vref calibration in efuse, Coding Scheme None
MAC: 24:0a:c4:62:64:14
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 5957.9 kbit/s)...
Hash of data verified.
Compressed 15856 bytes to 10276...
Wrote 15856 bytes (10276 compressed) at 0x00001000 in 0.1 seconds (effective 975.8 kbit/s)...
Hash of data verified.
Compressed 643600 bytes to 385424...
Wrote 643600 bytes (385424 compressed) at 0x00010000 in 5.5 seconds (effective 928.5 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1755.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

e3 DOIT ESP32 DEVKIT V1 on COM8
```

Slika 3.7 Prikaz obrade programa te prijensa na ESP32 mikrokontroler

Nakon unesenog programa, otvaranjem serijskog monitora direktno se može vidjeti stanje komunikacije ESP32 mikrokontrolera te opće informacije o povezanosti sa mrežom.

```
COM8
13:56:09.810 ->
13:56:09.857 ->
13:56:09.857 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
13:56:09.857 -> configsip: 0, SPIWP:0xee
13:56:09.857 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
13:56:09.857 -> mode:DIO, clock div:1
13:56:09.857 -> load:0x3fff0018,len:4
13:56:09.857 -> load:0x3fff001c,len:1044
13:56:09.857 -> load:0x40078000,len:8896
13:56:09.857 -> load:0x40080400,len:5816
13:56:09.857 -> entry 0x400806ac
13:56:10.090 -> Povezivanje na MATRIX
13:56:10.698 -> .
13:56:10.698 -> WiFi povezan.
13:56:10.698 -> IP adresa:
13:56:10.698 -> 192.168.1.103

 Autoscroll  Show timestamp
No line ending 115200 baud Clear output
```

Slika 3.8 Prikaz serijskog monitora porta COM8 nakon učitavanja programa

ESP32 mikrokontroler je uredno povezan na WiFi pristupnu mrežu te mu je pristup pregleda omogućen na IP adresi : **192.168.1.103**

3.6. Izgled pristupnog web sučelja

Prilikom upisa u web preglednik IP pristupnu adresu **192.168.1.103** otvara se prozor kao na donjoj slici. Omogućeno je upravljanje procesom na ponuđeni način. Odabirom „START AUTOMATSKI“ omogućava se pokretanje automatskog procesa kao prethodno definiranog u PLC programu. Odabirom od ostalih ponuđenih opcija bira se pojedini proces prema opis. Definirano je da uz svaki odabrani proces nije moguće birati sljedeći sve do drugog odabira, odnosno završetka odabranog.



Slika 3.9 Prikaz web sučelja prema IP pristupnoj točki

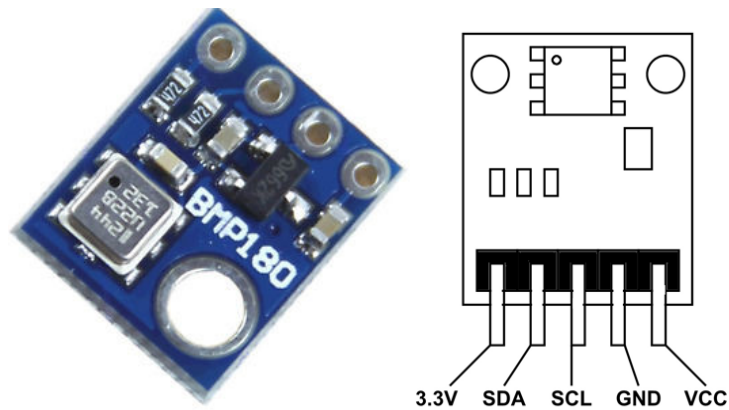


Slika 3.10 Prikaz web sučelja prema pojedinim uključenim procesima

3.7. Unaprjeđenje rada ESP32 pomoću BMP180

Kako bi se unaprijedio rad ESP32 mikrokontrolera sa odabirom procesa može se pomoću senzoričke BMP180 uvesti mogućnost kao primjer očitavanja trenutne temperature, tlaka i/ili nadmorske visine.

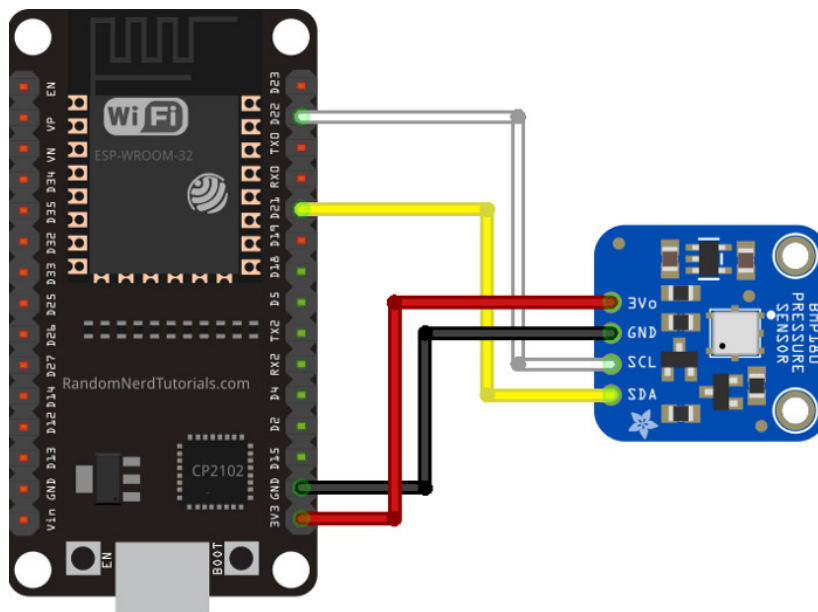
BMP180 je senzor za mjerenje barometarskog tlaka i temperature. Budući da se tlak mijenja s visinom, može se kao što je prethodno napomenuto, koristiti za mjerenje visine. Senzor je integriran s regulatorom od 3.3V što omogućuje direktnu naponsku povezanost s ESP32 mikrokontrolerom, ali je i usklađen s 5V naponskog stanja.



Slika 3.11 Prikaz BMP180 senzora te njegovih pinova [25] [26]

OSNOVNE TEHNIČKE SPECIFIKACIJE:

- Vcc: 3 do 5VDC
- Logika: usklađeno od 3 do 5V
- Raspon osjetljivosti tlaka: 300-1100 hPa (9000 m do -500 m nadmorske visine)
- Razlučivost do 0,03hPa / 0,25m
- Područje rada od -40 do + 85 ° C, točnost temperature od + -2 ° C
- Ova ploča / čip koristi I2C 7-bitnu adresu 0x77



Slika 3.12 Prikaz povezivanja ESP32 sa BMP180 [27]

Programskim rješenjem izradit će se mogućnost očitavanja temperature, tlaka i nadmorske visine prilikom uključanja pojedinog procesa. Nastavkom u postojeći programi potrebno je prije svega dodatno definirati biblioteke za rad BMP180 senzora.

```
// Ucitavanje biblioteke za rad BMP180 osjetnika
#include <Wire.h>
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;
```

Programski blok 11 Prikaz programa za definiranje biblioteka rada BMP180 osjetnika

`#include <Wire.h>` Ova biblioteka omogućuje komunikaciju s I2C / TWI uređajima. Kod BMP180 osjetnika postoji SDA (podatkovna linija) te SCL (satna linija).

`#include <Adafruit_BMP085.h>` Biblioteka dizajnirana posebno za rad Adafruit BMP085 ili BMP180 osjetnik.

```
void setup() {
  Serial.begin(115200);
  if (!bmp.begin())
  {
    Serial.println("Nije povezan sa BMP180, provjeriti ozicenje");
    while (1) {}
  }
}
```

Programski blok 12 Prikaz dodanog programa za povezivanje BMP180 osjetnika

Za unaprjeđenje sustava mjerenjem veličina potrebno je izraditi HTML tablicu. HTML tablica se definira oznakama `<table></table>`.

Stvaranjem retka moguće je oznakama `<tr></tr>`, a ćelija pomoću oznaka `<td></td>`

```
// Prikaz HTML obrazca za upis podataka u tabelu (temperatura, tlak i nadmorska
visina)
    client.println("</style></head><body><h2>UNAPRIJEDENJE SA OSJETNIKOM
BMP180</h2>");
    client.println("<table><tr><th>MJERENJA</th><th>VRIJEDNOSTI</th></tr>");
    client.println("<tr><td>Temp. u Celziju</td><td><span
class=\"sensor\">");
    client.println(bmp.readTemperature());
    client.println(" *C</span></td></tr>");
    client.println("<tr><td>Temp. u Fahrenheitima</td><td><span
class=\"sensor\">");
    client.println(1.8 * bmp.readTemperature() + 32);
    client.println(" *F</span></td></tr>");
    client.println("<tr><td>Tlak zraka</td><td><span class=\"sensor\">");
    client.println(bmp.readPressure() / 100.0F);
    client.println(" hPa</span></td></tr>");
    client.println("<tr><td>Priblizna nadmorska visina</td><td><span
class=\"sensor\">");
    client.println(bmp.readAltitude());
    client.println(" m</span></td></tr>");
    client.println("</body></html>");
```

Programski blok 13 Prikaz djela programa za ispis podataka u tablicu

Prema gornjoj slici stvoreno je zaglavlje tablice sa ćelijama „MJERENJA“ i „VRIJEDNOSTI“, zatim su definirani redci za prikaz očitavanja veličine pod opisom i njenom vrijednošću u tom trenutku.

Doradom programa za mjerenje veličina putem BMP180 upotpunjena je funkcionalnost i pristupačnost samog sustava.

Primjenom stilova CSS-a tablica će se po želji oblikovati.

```
// Prikaz CSS oblikovanja tabele za upis izmjerenih podataka
    client.println("<style>body { text-align: center; font-family:
\"Trebuchet MS\", Arial;}");
    client.println("<table { border-collapse: collapse; width:35%; margin-
left:auto; margin-right:auto; }");
    client.println("<th { padding: 12px; background-color: #0043af; color:
white; }");
    client.println("<tr { border: 1px solid #ddd; padding: 12px; }");
    client.println("<tr:hover { background-color: #bcbcbc; }");
    client.println("<td { border: none; padding: 12px; }");
    client.println(".sensor { color:white; font-weight: bold; background-
color: #bcbcbc; padding: 1px; }");
```

Programski blok 14 Prikaz djela programa za uređenje tablice upisa podataka

Nakon ažuriranog djela programa unaprijeđenog putem BMP180 osjetnika, može se ponovno prikazati simulacija tog sustava gdje se primjećuje dodatna tablica sa mjerenim veličinama.



Slika 3.13 Prikaz ažuriranog sučelja unaprijeđenog osjetnikom BMP180

MJERENJA	VRIJEDNOSTI
Temp. u Celziju	24.30 °C
Temp. u Fahrenheitima	75.74 °F
Tlak zraka	1003.82 hPa
Priblizna nadmorska visina	79.23 m

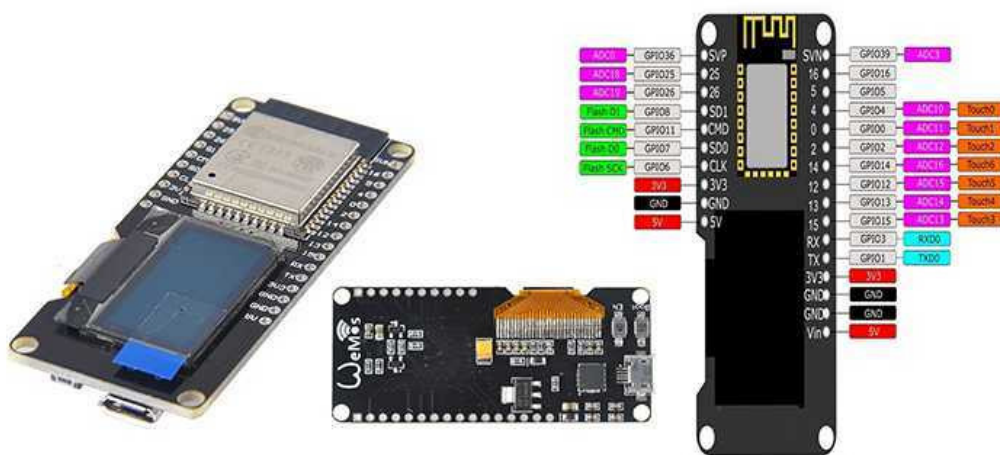
Slika 3.14 Prikaz tablice mjerenja osjetnikom BMP180

3.8. Karakteristike ESP32 mikrokontrolera

U izradi sustava upravljanja mikrokontrolerom ESP32 prema tehničkim specifikacijama postoje više tipova takvih mikrokontrolera. Izvedbe samostalnih modula za samostalno lemljenje ili oni sa razvojnim značajkama kontrolera tvrtke Sparkfun. U radu je korištena razvojna pločica tvrtke Adafruit, ali ipak, izravni konkurenti su kineski klonovi koji su jednako kompatibilni, jer koriste isti ESP32 čip proizveden strane Espressif Systems. Razlika je samo u nadograđenim

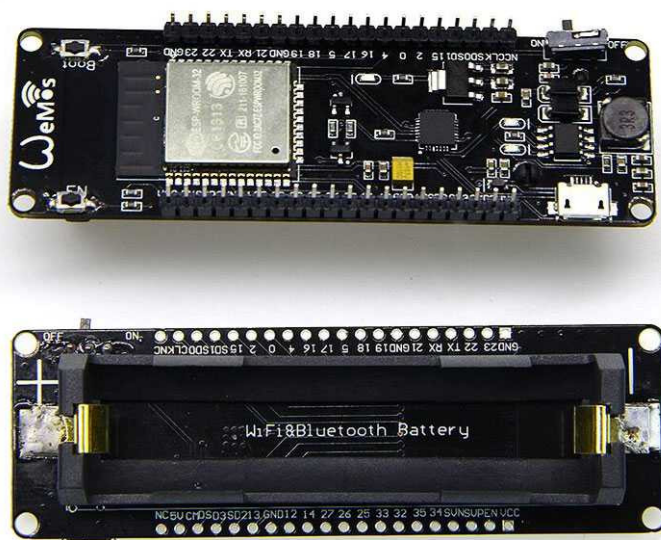
značajkama pojedinog mikrokontrolera te kvaliteta elektroničkih komponenata i preciznosti lemljenja.

Novi popularniji ESP32 razvojni sustavi s ugrađenim OLED zaslonom postaju sve zastupljeniji na tržištu. Razvojni inženjeri cijene takve razvojne ploče zbog lake transformacije u zapisivanje podataka ili djelovanja kao pristupnih WiFi točaka s prikazom podataka uživo. Svakako se mora obratiti pozornost na potrošnju energije koja je na takvim razvojnim pločama puno veća od običnih.



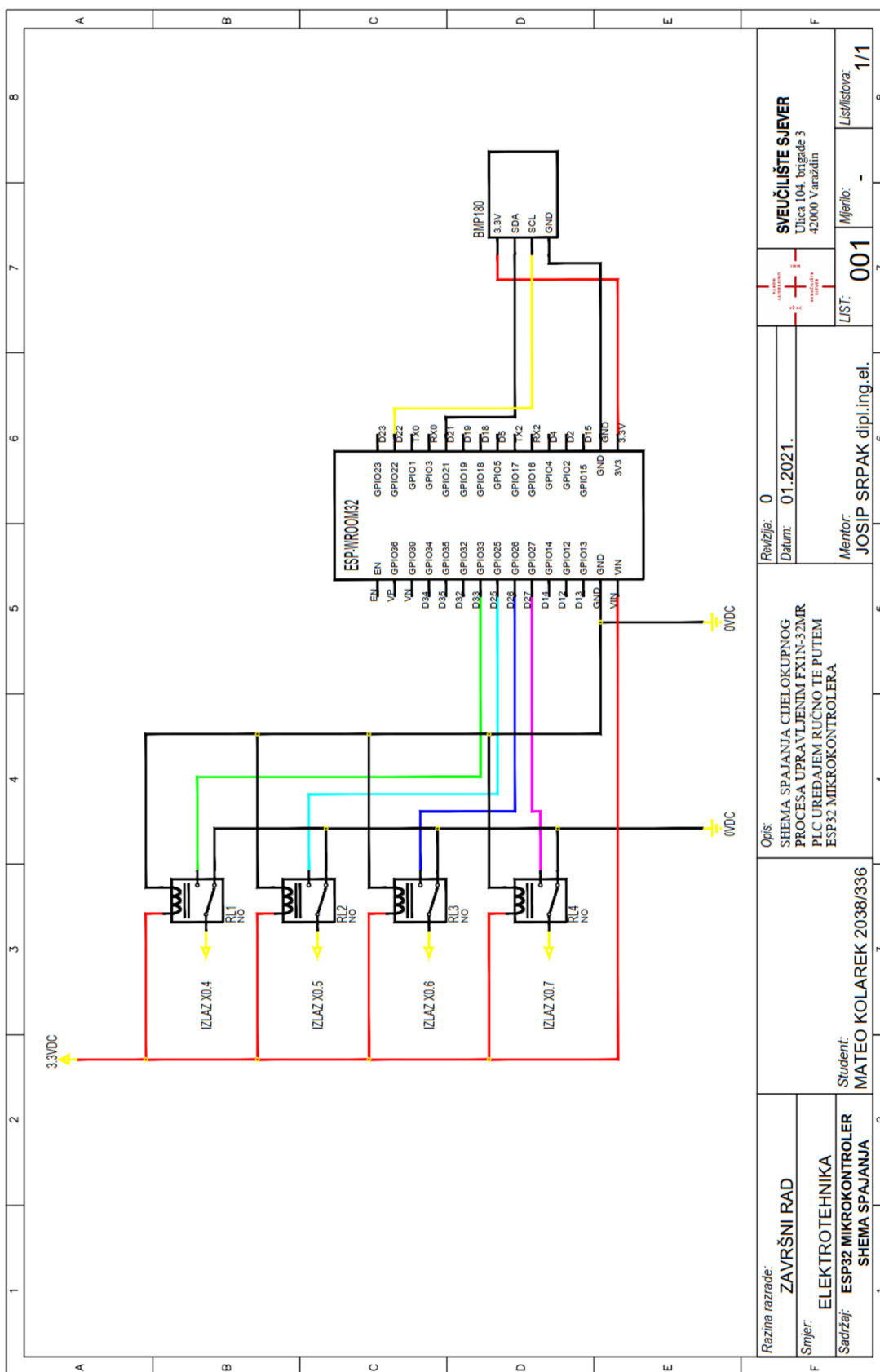
Slika 3.15 Prikaz ESP32 Makerfocus OLED razvojne ploče [30]

WEMOS tvrtka proizvodi ploču koju nazivaju LOLIN32 sa integrirani ESP32 čipom. Također postoji verzija koja ima ugrađen utor za bateriju kao prema donjoj slici.

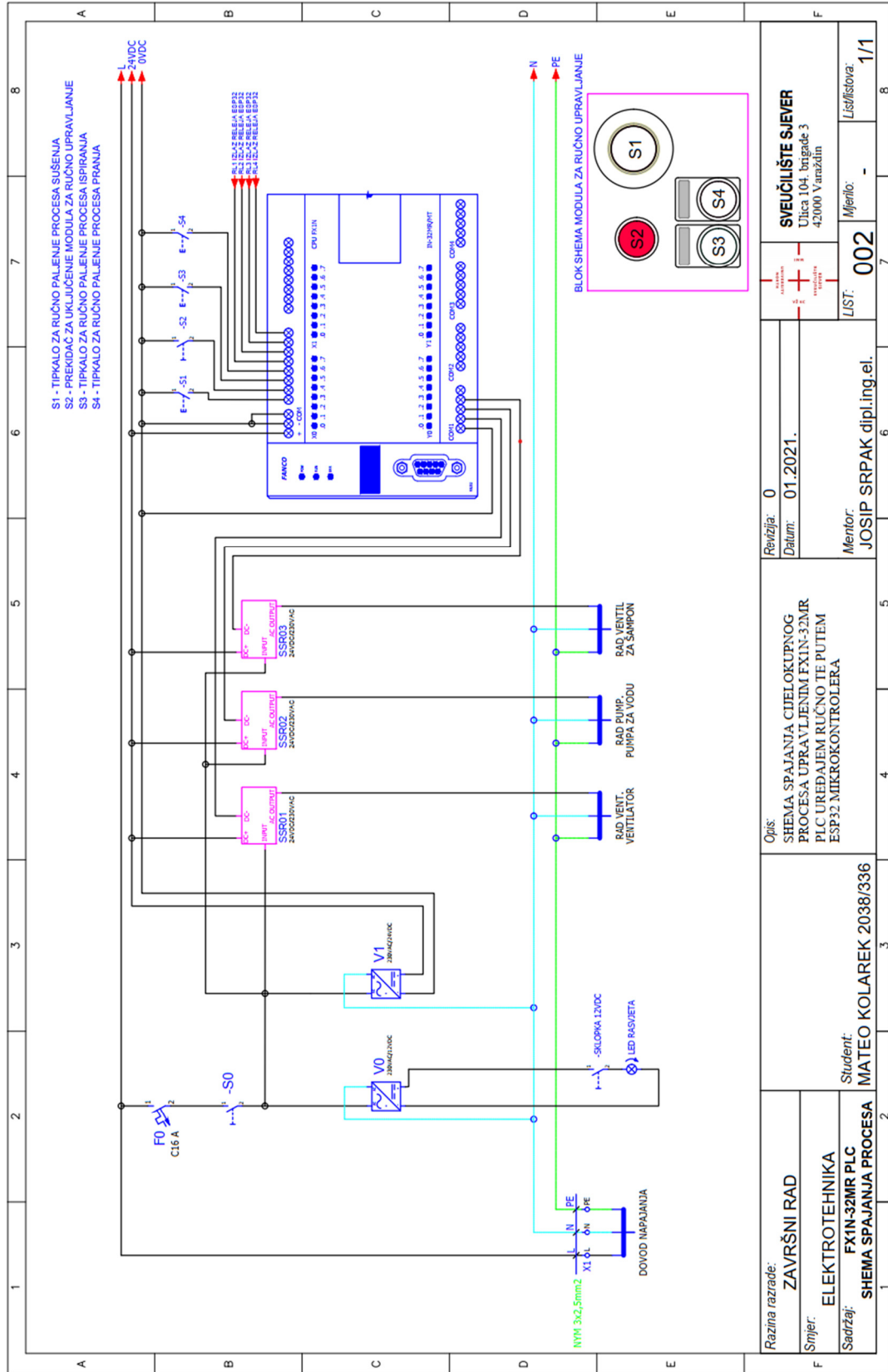


Slika 3.16 Prikaz WeMos ESP32 razvojnu ploču [30]

4. Shema sustava upravljanja



Slika 4.1 Shema spajanja ESP-WROOM-32 mikrokontrolera

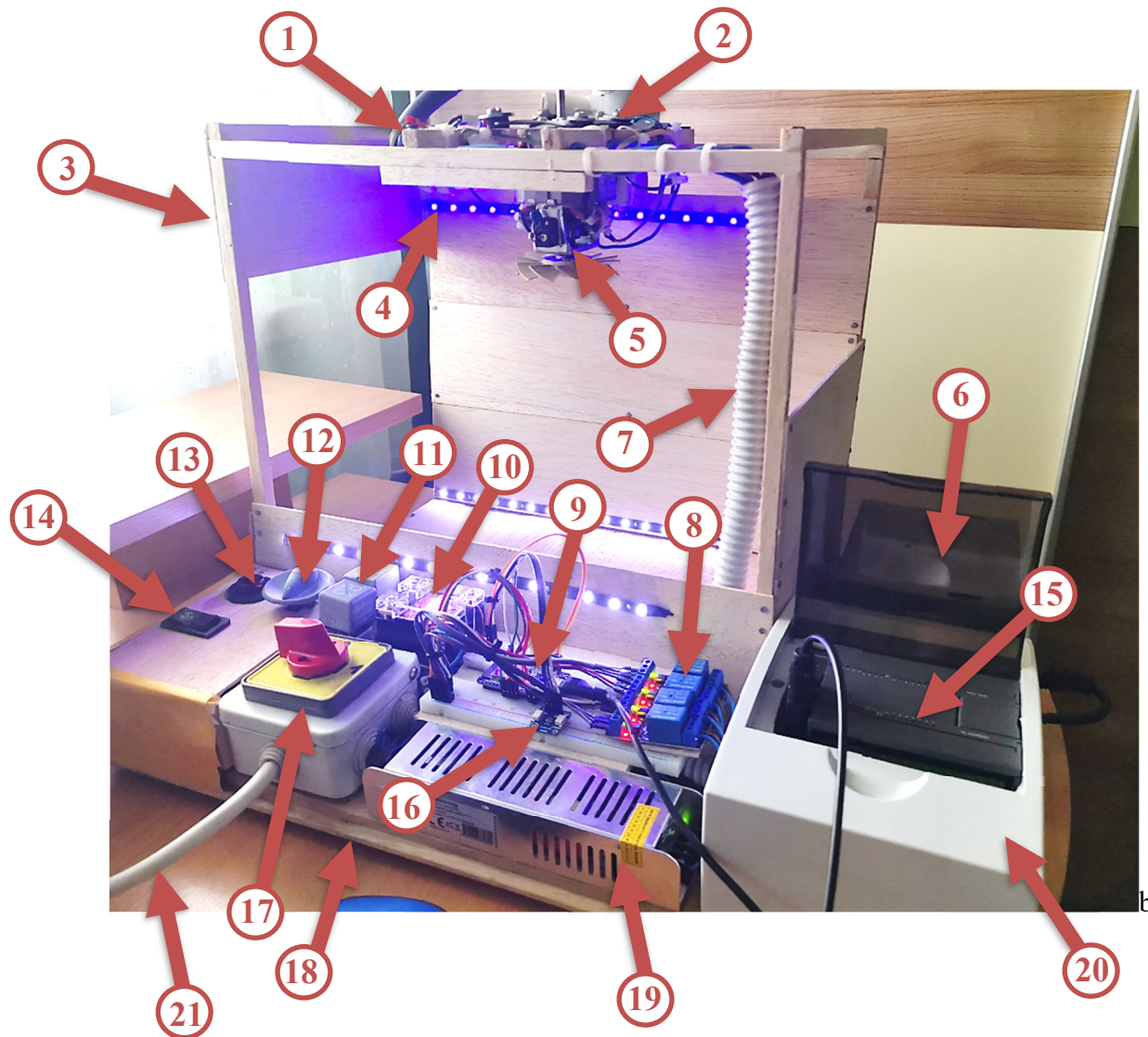


Slika 4.2 Shema spajanja FX1N-32MR PLC-a sa sustavom upravljanja

5. Simulacija sustava

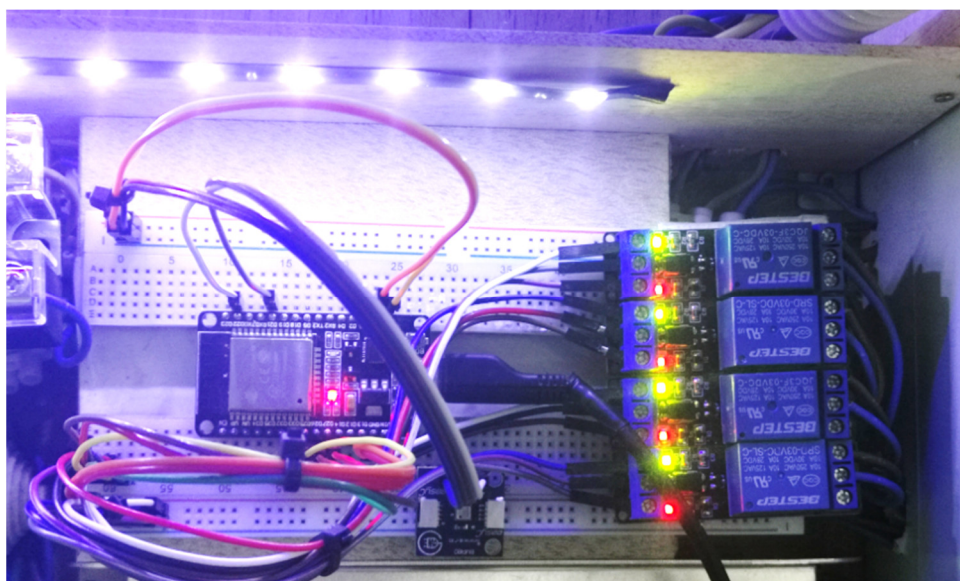
Simulacijom sustava primijenjen je koncept izrade procesa u laboratorijskim uvjetima rada koji jasno prikazuju rad pojedinih komponenata te vršenje nad istim.

5.1. Izgled sustava upravljanja



Slika 5.1 Prikaz simulacijskog izrađenog sustava

OPIS KOMPONENATA SUSTAVA	OZNAKA
ELEKTRO VENTIL ZA ŠAMPON, 230V, 50 Hz	1
VODENA PUMPA , 230V, 50 Hz	2
DRVENI OKVIR INPROVIZIRANE SAMOPOSLUŽNE PRAONICE	3
LED TRAKA OSVJETLJENJE 3 kom , 12VDC	4
VENTILATOR, 230V, 50 Hz	5
AC/DC ISPRAVLJAČ INPROVIZIRAN OD STAROG PC RAČUNALA, 12VDC	6
INSTALACIJSKA CIJEV DWP20mm – smješteni vodovi za komponente	7
ARDUINO RELEJI, upravljačka jedinica 3VDC, 4kom	8
ESP32 MIKROKONTROLER	9
SSR RELEJ (INPUT 4-32VDC ; OUTPUT 0-480VAC)	10
SSR RELEJ (INPUT 4-32VDC ; OUTPUT 0-480VAC) 2kom	11
TIPKALO ZA PALJENJE RUČNO VENTILATORA	12
SKLOPKA ZA UKLJUČENJE MODULARNOG RUČNOG REŽIMA RADA	13
TIPKALA (LIJEV-ISPIRANJE; DESNO-PRANJE)	14
FX1N PLC UREĐAJ – SMJEŠTEN U PVC RAZVODNOM ORMARIĆU	15
BMP180 OSJETNIK	16
GREBENASTA SKLOPKA ZA UKLJUČENJE CIJELOKUPNOG SUSTAVA	17
DRVENA PODLOGA	18
AC/DC ISPRAVLJAČ 24VDC	19
NADGRADNI PVC RAZVODNI ORMAR	20
NYM-PGP 3x2,5mm ² KABEL , 2m	21



Slika 5.2 Prikaz ESP32 te BMP180 senzoričke sa Arduino relejima

6. Zaključak

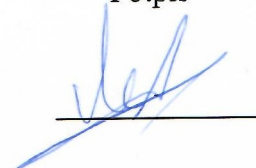
Svaki automatizirani sustav je moguće unaprijediti sa različitim konceptima upravljanja procesa. U završnom radu unaprijeđen je sustav upravljanja PLC-a putem ESP32 mikrokontrolera te dodatnim upravljanjem i vizualizacijom putem WinCC SCADA sustava. Razvojem takvih sustava omogućuje se brže i efikasnije pristupanje procesu ili prikupljanje podataka. U današnje vrijeme putem 5G mreža moguće je gotovo u realnom vremenu upravljanje procesom na udaljenom poslužitelju. PLC sustavom upravljanja prikazana je jednostavnost primjene i programiranja te implementacija sustava sa komponentima vezanih uz upravljanje procesa kroz primjer samoposlužne praonice. Osjetnicima se može simulirati pristup očitavanja različitih varijabli kojima se implementira sljedeća faza procesa. Također uz nastali kvar, svaki dio je moguće analizirati u najkraćem vremenskom periodu i time otkloniti poteškoće za daljnji rad. Pouzdanost takvih sustava ovisi o kvaliteti samih komponenata i programskog rješenja.

Povezivanjem FX1N-32MR PLC uređaja s ESP32 mikrokontrolerom omogućuje se upravljanje procesom samoposlužne praonice bežično putem WiFi mreže. Direktnom metodom povezivanja s relejima djeluje se na ulaz PLC-a te time se pokreće faza procesa prema programu PLC-a izrađenim u MELSOFT okruženju. Povezivanjem WinCC SCADA sustava putem MX OPC servera s FX1N-32MR PLC-om omogućena je komunikacija te izrada vizualizacije procesa kroz definirane tagove u MX Configurator-u. Takav pristup prikazuje povezanost dvaju različitih okruženja.

Razvojem RS-485 komunikacije sa ESP32 mikrokontrolerom došlo bi do poboljšanja samog upravljanja procesa. RS-485 komunikacijom moguće je putem ESP32 mikrokontrolera direktno povezivanje na PLC uređaj pomoću dva vodiča (A i B) i upravljati putem adresnih lokacija. Time bi se povećala učinkovitost, brzina slanja i primanja podataka, jednostavnije upravljanje i zaključno tome relejno upravljanje ne bi bilo potrebno.

U Varaždinu, 15.02.2021.

Potpis



7. Literatura

- [1] <https://int.rsdelivers.com/product/siemens/6es7214-1ad23-0xb0/siemens-s7-200-plc-cpu-14-inputs-10-outputs/4886713> dostupno 2021.
- [2] <https://blog.se.com/industrial-software/2020/02/03/why-are-plc-scada-and-hmi-perfect-for-industrial-automation/> dostupno 2021.
- [3] <https://library.automationdirect.com/history-of-the-plc/> dostupno 2021.
- [4] <https://www.electrical4u.com/programmable-logic-controllers/> dostupno 2021.
- [5] <http://spvp.zesoi.fer.hr/predavanja/skripta/rs232.pdf> dostupno 2021.
- [6] <http://spvp.zesoi.fer.hr/seminari/2003/sipic/RS-485.htm> dostupno 2021.
- [7] <https://www.aliexpress.com/item/4000707144836.html> dostupno 2021.
- [8] <https://www.inductiveautomation.com/resources/article/what-is-scada> dostupno 2021.
- [9] <https://www.hw-group.com/software/hercules-setup-utility> dostupno 2021.
- [10] <https://dcandcn.blogspot.com/2019/11/what-is-enqack.html> dostupno 2021.
- [11] <http://www.physics.udel.edu/~watson/scen103/ascii.html> dostupno 2021.
- [12] <https://circuits4you.com/2018/12/31/esp32-devkit-esp32-wroom-gpio-pinout/> dostupno 2021.
- [13] <https://program-plc.blogspot.com/2015/11/plc-modbus-master-arduino-modbus-slave.html> dostupno 2021.
- [14] <https://core-electronics.com.au/ttl-uart-to-rs485-converter-module.html> dostupno 2021.
- [15] <https://eu3a.mitsubishielectric.com/fa/en/dl/9886/MXOPCConfiguratorQuickStart.pdf> dostupno 2021.
- [16] <http://www.bizkit.ru/en/2019/02/21/12563/> dostupno 2021.
- [17] <https://diy0t.com/relay-tutorial-for-arduino-and-esp8266/> dostupno 2021.
- [18] https://en.wikipedia.org/wiki/Client%E2%80%93server_model dostupno 2021.
- [19] https://en.wikipedia.org/wiki/End-of-Text_character dostupno 2021.
- [20] <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/server-examples.html> dostupno 2021.
- [21] <https://www.arduino.cc/reference/en/language/functions/communication/serial/begin/> dostupno 2021.
- [22] <https://www.arduino.cc/en/Reference/WiFiServerAvailable> dostupno 2021.
- [23] <https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/> dostupno 2021.
- [24] <https://www.adafruit.com/product/1603> dostupno 2021.
- [25] <https://diyprojects.io/bmp180-atmospheric-pressure-temperature-sensor-create-barometer/#.YBHoEmhKiUk> dostupno 2021.
- [26] <https://components101.com/sensors/bmp180-atmospheric-pressure-sensor> dostupno 2021.
- [27] <https://medium.com/@pavitaardhani/a-journey-of-embedded-system-barometric-pressure-sensor-with-esp32-ft-bmp180-3345a024c846> dostupno 2021.
- [28] <https://www.arduino.cc/en/reference/wire> dostupno 2021.
- [29] <https://github.com/adafruit/Adafruit-BMP085-Library> dostupno 2021
- [30] <https://www.geekstips.com/esp32-review-idf-programming-tutorial/> dostupno 2021
- [31] <https://randomnerdtutorials.com/esp32-web-server-with-bme280-mini-weather-station/> dostupno 2021

[32] <https://www.instructables.com/Interfacing-BMP-180-Barometric-Pressure-Sensor-and/>
dostupno 2021

Popis slika

Slika 1.1 Prikaz PLC uređaja tip Siemens Simatic S7-200 [1]	10
Slika 1.2 Ljestvičasti prikaz programiranja PLC-a [3].....	11
Slika 1.3 Prikaz blok dijagrama PLC uređaja [4].....	13
Slika 1.4 Prikaz povezivanja digitalne komunikacije standarda RS-232 [5]	14
Slika 1.5 Prikaz PLC programiranja Ladder dijagramom [4]	15
Slika 1.6 Prikaz PLC programiranja FBD-om [4].....	16
Slika 1.7 Primjer programiranja Ladder-om (lijevo) te FBD-om(desno) [4].....	16
Slika 1.8 Prikaz blok sheme FX1N-32MR PLC-a prema proizvođaču.....	17
Slika 1.9 Prikaz izgleda FX1N-32MR PLC-a [7].....	17
Slika 1.10 Prikaz izgleda MELSOFT programskog sučelja sa izradom novog projekta	18
Slika 1.11 Prikaz radne trake sa osnovnim funkcijama MELSOFT programa	18
Slika 1.12 Prikaz izgleda konfiguracije povezivanja sa FX1N-32MR.....	19
Slika 1.13 Prikaz obavijesti uspješnog povezivanja sa FX1N-32MR	20
Slika 1.14 Tablični prikaz odabranih broja ulaza PLC-a sa opisom	20
Slika 1.15 Tablični prikaz odabranih broja izlaza PLC-a sa opisom.....	20
Slika 1.16 Prikaz otvorenog (lijevo) i zatvorenog (desno) kontakta	21
Slika 1.17 Prikaz otvorenog kontakta sa pozitivnim(rastućim) impulsnim bridom (lijevo) i otvorenog kontakta sa negativnim(padajućim) impulsnim bridom (desno).....	21
Slika 1.18 Prikaz zavojnice releja – izlaz PLC-a.....	22
Slika 1.19 Prikaz definiranja timera T0 – oznaka ; K100 – trajanje od 10s.....	22
Slika 1.20 Prikaz definiranja funkcija SET i RESET izlaza Y0.0.....	22
Slika 1.21 Prikaz izrade PLC programa za rad ručnog instaliranog modula.....	22
Slika 1.22 Prikaz djela programa sa timerom od 30s za uključenje izlaza Y0.0 (rad ventilatora) i Y0.1(rad pumpe).....	23
Slika 1.23 Prikaz djela programa sa timerom od 30s za uključenje izlaza Y0.2 (rad ventila)	23
Slika 1.24 Prikaz djela programa za reset pomoćne varijable preko timera.....	24
Slika 1.25 Prikaz djela programa kod isključenja ulazne sklopke X0.1.....	24
Slika 1.26 Prikaz djela programa kod upravljanja putem mikrokontrolera.....	25
Slika 1.27 Prikaz djela programa za uključenje timera automatskog režima rada	25
Slika 1.28 Prikaz djela programa za pozivanje rada procesa unutar vremenskog intervala.....	26
Slika 1.29 Prikaz simulacije programa kod automatskog rada – X0.7 je isključen	26
Slika 1.30 Prikaz simulacije programa kod automatskog rada – X0.7 je uključen	27
Slika 1.31 Prikaz simulacije programa kod odbrojavanja rada timera T3.....	27

Slika 1.32 Prikaz simulacije programa kod rada ventilatora Y0.0	28
Slika 1.33 Prikaz simulacije programa kod rada timera T3.....	28
Slika 2.1 Prikaz prvih izgleda „monolitnih“ SCADA sustava [8].....	29
Slika 2.2 Prikaz hw-group stranice za preuzimanje terminala Hercules [9].....	30
Slika 2.3 Prikaz Hercules terminala u okruženju serijske komunikacije	31
Slika 2.4 Prikaz COM porta kod upravitelja uređaja.....	31
Slika 2.5 Prikaz Hercules terminala kod slanja HEX adrese 05.....	32
Slika 2.6 Prikaz odnosa okvira ENQ/ACK [10].....	32
Slika 2.7 Prikaz uključivanja i isključivanja izlaza PLC-a Y0.0 putem terminala.....	33
Slika 2.8 Prikaz STX i ETX iz ASCII tablice [11].....	33
Slika 2.9 Prikaz novog projekta u programu SIMATIC WinCC Explorer	34
Slika 2.10 Prikaz izrađenih slika vizualizacije	35
Slika 2.11 Prikaz definiranog PLC uređaja FX1N u MX OPC serveru	35
Slika 2.12 Prikaz napravljenih grupa tagova u MX OPC serveru FX1N PLC-a.....	36
Slika 2.13 Prikaz postavke tag-a kod definiranja	36
Slika 2.14 Prikaz definiranih tag-ova a) izlaza i b) ulaza FX1N PLC-a.....	37
Slika 2.15 Prikaz simulacije OPC servera te kompatibilnost sa programom PLC-a.....	38
Slika 2.16 Prikaz konfiguratora OPC drivera u WinCC-u	39
Slika 2.17 Prikaz izlaznih varijabli tagova unutar pretraživača OPC servera	39
Slika 2.18 Prikaz ulaznih varijabli tagova unutar pretraživača OPC servera.....	40
Slika 2.19 Prikaz FX1N tag liste unutar WinCC-a.....	40
Slika 2.20 Prikaz izrade naslovne slike vizualizacije WinCC-a.....	41
Slika 2.21 Prikaz konfiguracijskog prozora za definiranje tipke.....	41
Slika 2.22 Prikaz definiranja C akcije na klik	42
Slika 2.23 Prikaz C programskog koda za izlaz iz programa na klik.....	42
Slika 2.24 Prikaz određivanja događaja putem dinamičkog prozora	43
Slika 2.25 Prikaz slike stanja upravljanja procesa.....	44
Slika 2.26 Prikaz definiranja rada tipke te prikaz tipke.....	44
Slika 2.27 Prikaz definiranja događaja kod vizualizacije procesa.....	45
Slika 2.28 Prikaz definiranja događaja putem dinamičkog prozora	46
Slika 2.29 Prikaz slike vizualizacije procesa.....	46
Slika 2.30 Prikaz definiranja parametra trenda grafičkog prikaza	47
Slika 2.31 Prikaz definiranja parametra trenda tabličnog prikaza	48
Slika 2.32 Prikaz slike analize upravljanja	48
Slika 2.33 Prikaz naslovne slike prilikom simulacije.....	49

Slika 2.34 Prikaz stanja upravljanja prilikom simulacije	49
Slika 2.35 Prikaz stanja upravljanja prilikom simulacije automatskom rada.....	50
Slika 2.36 Prikaz stanja upravljanja prilikom simulacije rada ručnog modula	50
Slika 2.37 Prikaz slike procesa prilikom mirovanja u simulaciji	51
Slika 2.38 Prikaz slike procesa prilikom simulacije rada pumpe procesa ispiranja	51
Slika 2.39 Prikaz slike procesa pranja	52
Slika 2.40 Prikaz slike procesa sušenja	52
Slika 2.41 Prikaz analize upravljanja prilikom simulacije iz procesa ispiranja u proces pranja ...	53
Slika 3.1 Prikaz pinova ESP32 mikrokontrolera [12]	54
Slika 3.2 Prikaz ESP32 mikrokontrolera [12]	54
Slika 3.3 Prikaz TTL-RS485 pretvarača [14].....	55
Slika 3.4 Prikaz povezivanja ESP32 mikrokontrolera sa TTL-RS485 [16].....	56
Slika 3.5 Prikaz povezivanja ESP32 mikrokontrolera sa Arduino relejem (3.3-5V) [17]	56
Slika 3.6 Prikaz odnosa klijent-poslužitelj [18].....	64
Slika 3.7 Prikaz obrade programa te prijenosa na ESP32 mikrokontroler	65
Slika 3.8 Prikaz serijskog monitora porta COM8 nakon učitavanja programa	65
Slika 3.9 Prikaz web sučelja prema IP pristupnoj točki	66
Slika 3.10 Prikaz web sučelja prema pojedinim uključenim procesima	67
Slika 3.11 Prikaz BMP180 senzora te njegovih pinova [25] [26]	68
Slika 3.12 Prikaz povezivanja ESP32 sa BMP180 [27]	69
Slika 3.13 Prikaz ažuriranog sučelja unaprijednog osjetnikom BMP180	71
Slika 3.14 Prikaz tablice mjerenja osjetnikom BMP180	71
Slika 3.15 Prikaz ESP32 Makerfocus OLED razvojne ploče [30]	72
Slika 3.16 Prikaz WeMos ESP32 razvojnu ploču [30].....	72
Slika 4.1 Shema spajanja ESP-WROOM-32 mikrokontrolera.....	73
Slika 4.2 Shema spajanja FX1N-32MR PLC-a sa sustavom upravljanja.....	74
Slika 5.1 Prikaz simulacijskog izrađenog sustava	75
Slika 5.2 Prikaz ESP32 te BMP180 senzoričke sa Arduino relejima	76

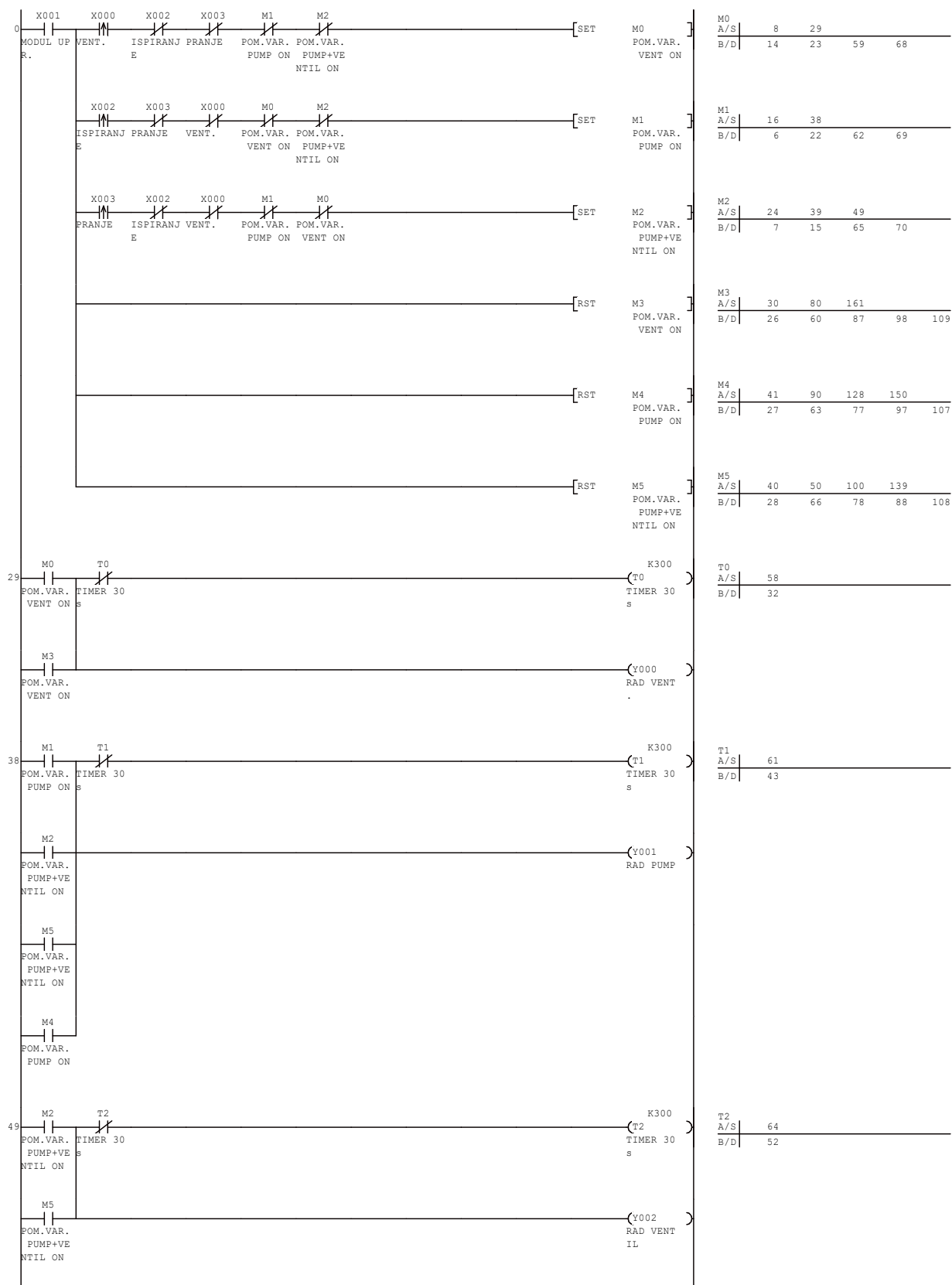
Programski blok

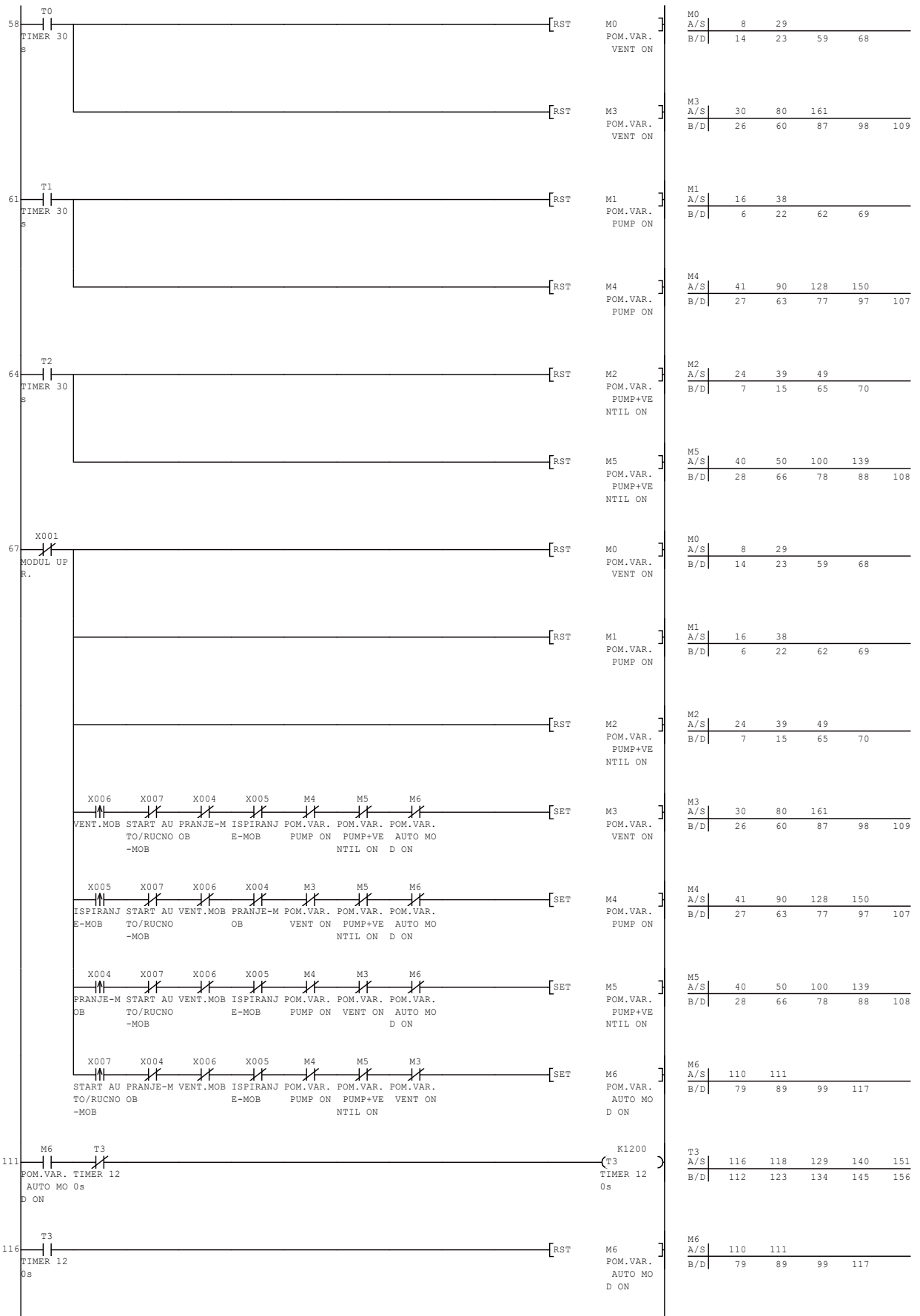
Programski blok 1 Prikaz programa za definiranje biblioteke, WiFi pristupa te postavljane porta	57
Programski blok 2 Prikaz programa za definiranje izlaznih pinova.....	58
Programski blok 3 Prikaz programa za povezivanje na mrežu te ispis podataka	58
Programski blok 4 Prikaz programa za slanje i primanje podataka između klijenta i poslužitelja	59
Programski blok 5 Prikaz programa za ispis stanja pinova prilikom promjene stanja.....	60
Programski blok 6 Prikaz programa za ispis oblika HTML stranice	60
Programski blok 7 Prikaz osnovne strukture HTML obrasca	60
Programski blok 8 Prikaz definiranog naslova web stranice.....	61
Programski blok 9 Prikaz programa za definiranje odabira rada procesa korisničkog sučelja	62
Programski blok 10 Prikaz programa za definiranje HTTP odgovara – Kraj programa	63
Programski blok 11 Prikaz programa za definiranje biblioteka rada BMP180 osjetnika.....	69
Programski blok 12 Prikaz dodanog programa za povezivanje BMP180 osjetnika.....	69
Programski blok 13 Prikaz djela programa za ispis podataka u tablicu	70
Programski blok 14 Prikaz djela programa za uređenje tablice upisa podataka.....	70

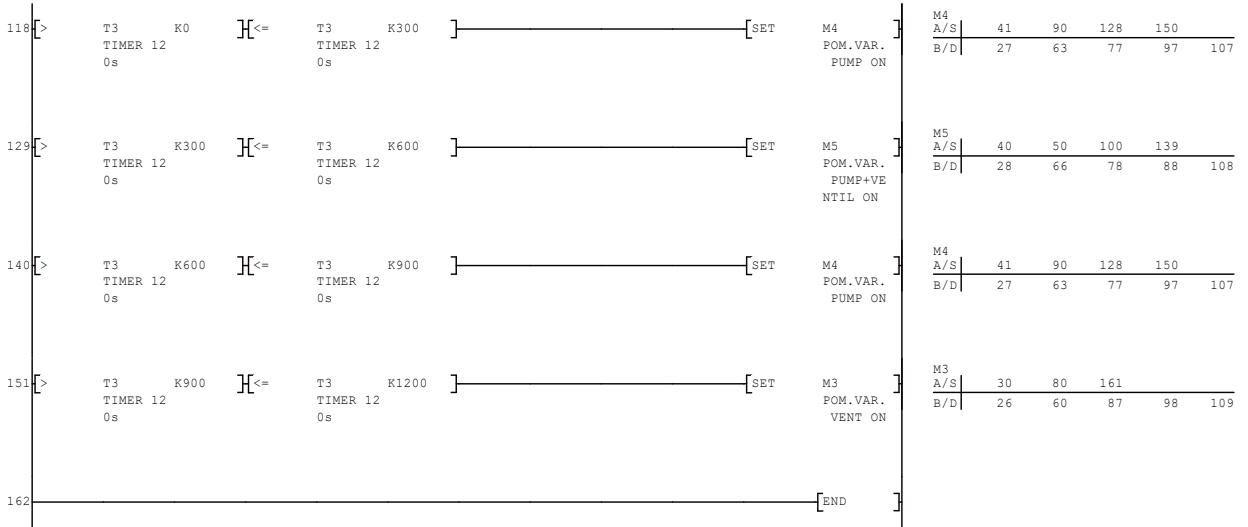
Prilozi

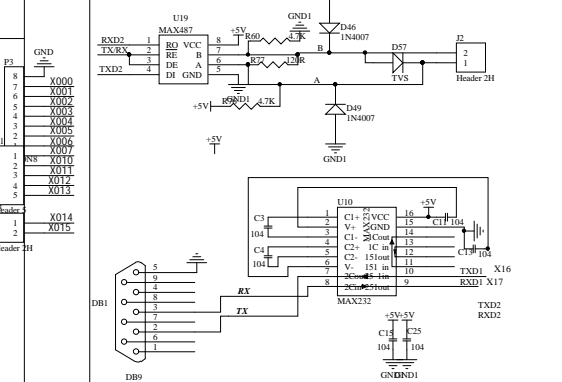
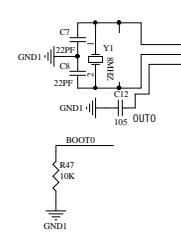
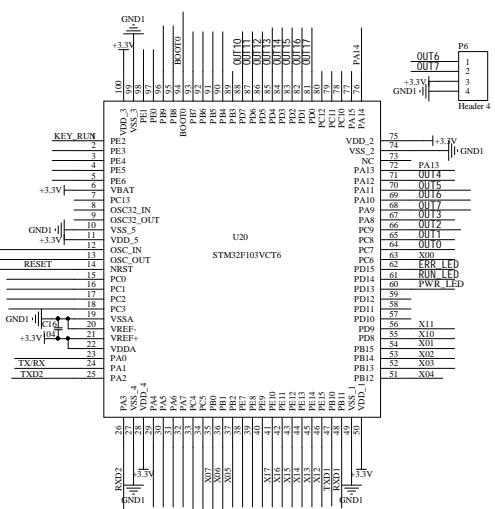
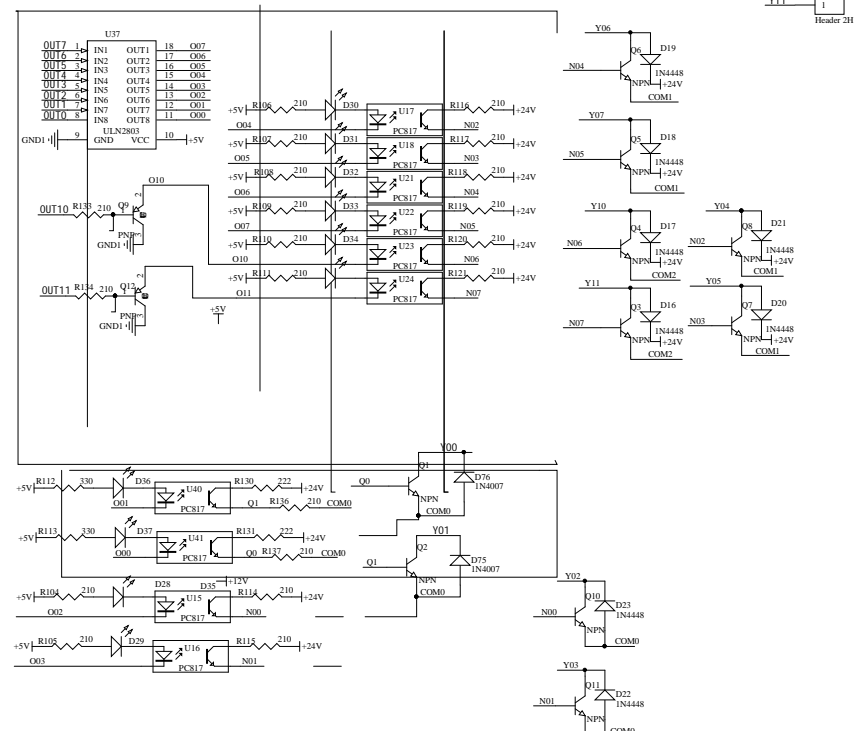
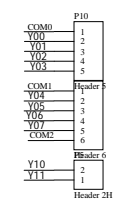
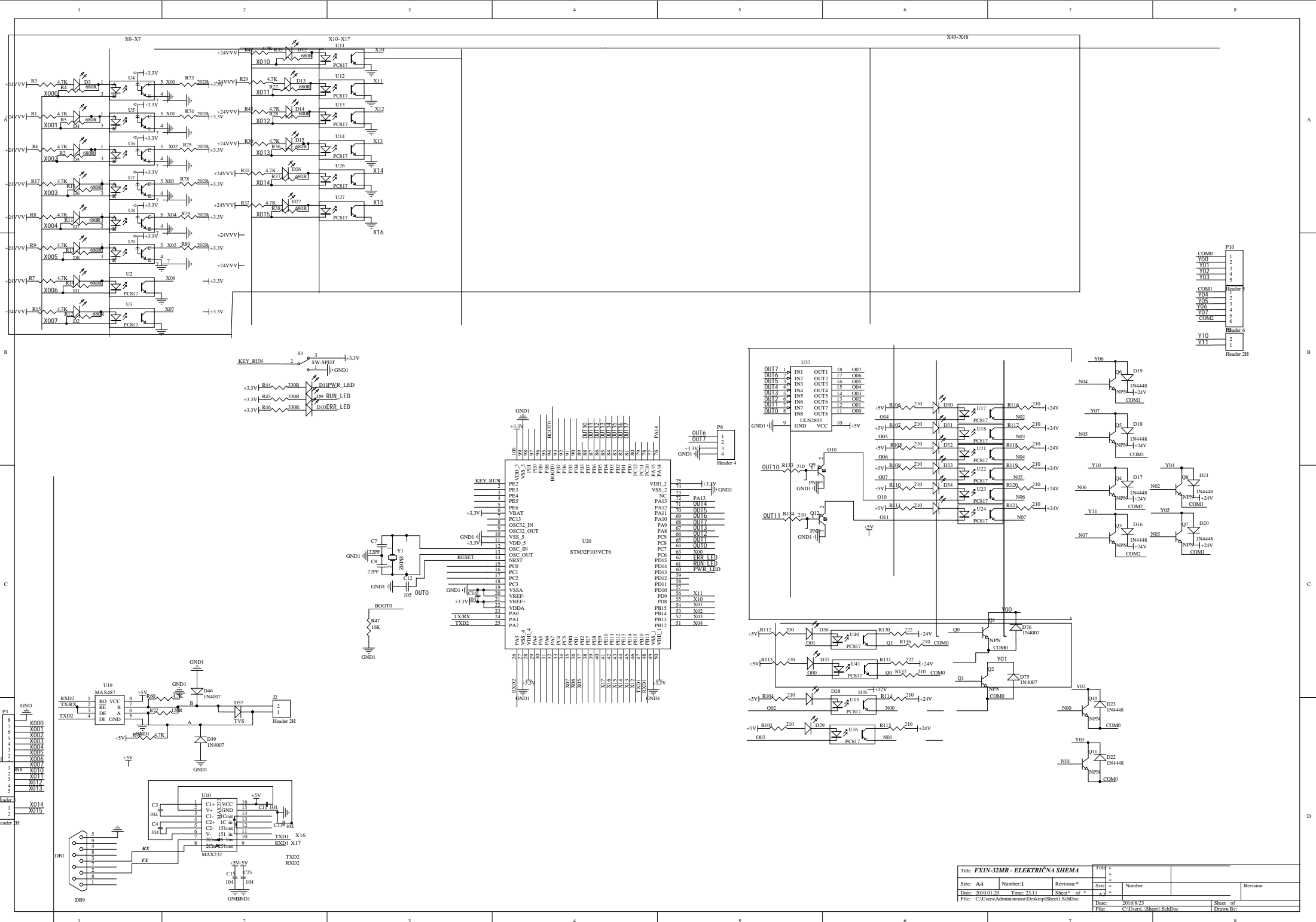
- 01 *FX1N-32MR – Ljestvičasti dijagram programa samoposlužne praonice*
- 02 *FX1N-32MR – Električna shema PLC uređaja*
- 03 *ESP-WROOM-32 – Električna shema ESP32 razvojnog okruženja*

FX1N-32MR–Ljestvičasti dijagram programa samoposlužne praonice









Title: FXIN-32MR - ELEKTRICNA SCHEMA			1/10		
Size: A4	Number: 1	Revision: *			
Date: 2010.01.30	Time: 22.11	Sheet: * of *			
File: C:\Users\Administrator\Desktop\Sheet1_SchDoc			Date: 2016/8/23	Sheet: 1	Revision:
			File: C:\Users\Administrator\Desktop\Sheet1_SchDoc	Drawn By:	

Schematics

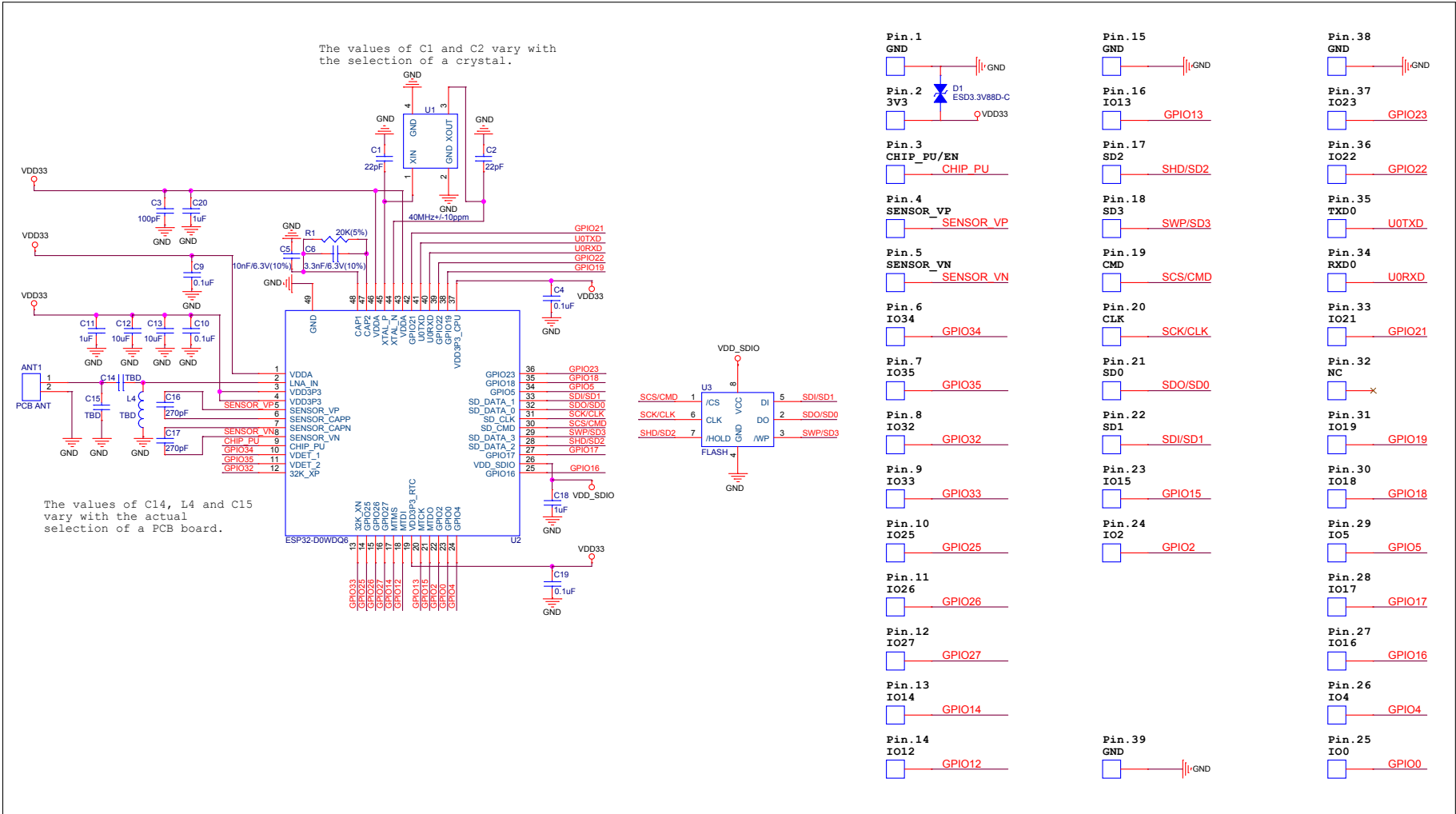


Figure 3: ESP32-WROOM-32 Schematics

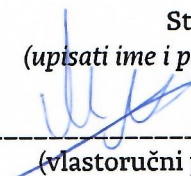
IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, **Mateo Kolarek** (*ime i prezime*) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor završnog rada pod naslovom:

Unaprjeđenje PLC sustava upravljanja primjenom mikrokontrolera (*upisati naslov*) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student:
(*upisati ime i prezime*)



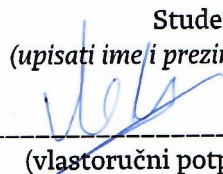
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, **Mateo Kolarek** (*ime i prezime*) neopozivo izjavljujem da sam suglasan s javnom objavom završnog rada pod naslovom:

Unaprjeđenje PLC sustava upravljanja primjenom mikrokontrolera (*upisati naslov*) čiji sam autor.

Student:
(*upisati ime i prezime*)



(vlastoručni potpis)