

# Daljinsko upravljanje dijelom procesa proizvodnje piva

---

**Blažić, Antun**

**Master's thesis / Diplomski rad**

**2025**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:168:591139>

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

*Download date / Datum preuzimanja:* **2025-03-29**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 118

**DALJINSKO UPRAVLJANJE DIJELOM PROCESA  
PROIZVODNJE PIVA**

Antun Blažić

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 118

**DALJINSKO UPRAVLJANJE DIJELOM PROCESA  
PROIZVODNJE PIVA**

Antun Blažić

Zagreb, veljača 2025.

## DIPLOMSKI ZADATAK br. 118

Pristupnik: **Antun Blažić (0036519293)**  
Studij: Informacijska i komunikacijska tehnologija  
Profil: Automatika i robotika  
Mentor: izv. prof. dr. sc. Vinko Lešić

Zadatak: **Daljinsko upravljanje dijelom procesa proizvodnje piva**

### Opis zadatka:

Automatizirani sustav za prihvata sirovina početni je korak u procesu proizvodnje piva. Prihvata se obavlja prijenosom iz kamiona pomoću trakastih transportera te dizalima prenosi do silosa uz otklanjanje nečistoća. U radu je potrebno uspostaviti udaljeni upravljački sustav kao SCADA platformu za prikupljanje, pohranu i prikaz podataka na udaljenom poslužiteljskom računalu koji sa sensorima i aktuatorima različitih proizvođača komunicira putem standardiziranih protokola i baze podataka. Implementaciju opisanog zadatka potrebno je provesti na odabranoj stvarnoj komercijalnoj platformi i odabranom segmentu za prihvata i filtriranje sirovina kao dijela procesa proizvodnje piva.

Rok za predaju rada: 14. veljače 2025.

*Zahvaljujem se profesoru Vinku Lešiću na pomoći i suradnji tijekom diplomskog studija.*

*Također, veliku zahvalu upućujem tvrtki Montelektro d.o.o na pomoći pri izradi  
diplomskog rada.*

*Na kraju, posebna zahvala cijeloj obitelji na podršci i motivaciji tijekom studija, a  
posebno roditeljima, sestrama i djevojci.*

# Sadržaj

<b>1. Uvod</b>	<b>3</b>
<b>2. Brewmaxx sustav</b>	<b>4</b>
2.1. Brewmaxx arhitektura	5
2.1.1. Server	5
2.1.2. Operaterska stanica	6
2.1.3. Inženjerska stanica	7
2.1.4. Programirljivi logički kontroler (PLC)	8
2.2. ISA-88: Standard za šaržnu proizvodnju	8
2.3. Proceduralno upravljanje sustavom	9
2.3.1. Sekvenca	10
2.3.2. Procesne operacije	11
2.3.3. Parametri procesa	11
2.3.4. Poruke	12
2.4. Upravljačke komponente sustava	13
2.4.1. Upravljanje upravljačkim komponentama	15
2.5. Vizualizacijsko sučelje	17
<b>3. Automatizacija sustava za pranje objekata</b>	<b>18</b>
3.1. Sustav za pranje objekata	18
3.2. Sekvenca tankova	20
3.2.1. Korak 1: Inicijalizacija (Initialising)	20
3.2.2. Korak 2: Punjenje (Filling)	22
3.2.3. Korak 3: Preparacija (Preparation)	22
3.2.4. Korak 4: Odabir linije (Line selection)	24

3.2.5.	Korak 5: Predrad linije (Preun line)	27
3.2.6.	Korak 6: Zagrijavanje (Heating)	29
3.2.7.	Korak 7: Poslijeradnja linije (Postrun line)	30
3.2.8.	Korak 8: Ispiranje linije (Flush line)	30
3.2.9.	Korak 9: Oslobađanje linije (Release line)	31
3.2.10.	Korak 10: Servis (Service)	31
3.2.11.	Korak 11: Kraj (End)	31
3.3.	Sekvenca linije	32
3.3.1.	Korak 1: Inicijalizacija (Initialising)	32
3.3.2.	Korak 2: Servis (Service)	32
3.3.3.	Korak 3: Kraj (End)	32
3.4.	Uloga kiseline i lužine u pranju DAW linije i filtera	33
3.4.1.	DAW linija	33
3.4.2.	Filteri	33
3.4.3.	Uloga lužine u pranju	33
3.4.4.	Uloga kiseline u pranju	34
3.4.5.	Kada se koriste kiselina i lužina	34
<b>4.</b>	<b>Prikupljanje i obrada podataka</b>	<b>35</b>
4.1.	Dijagnostika na temelju prikupljenih podataka	36
4.1.1.	Status komponente	37
4.1.2.	Vrijeme odziva aktuatora	43
4.2.	Predikcija servisa industrijskih ventila	55
4.3.	Cjelokupna arhitektura sustava	59
<b>5.</b>	<b>Zaključak</b>	<b>60</b>
	<b>Literatura</b>	<b>62</b>
	<b>Sažetak</b>	<b>63</b>
	<b>Abstract</b>	<b>64</b>

# 1. Uvod

Proizvodnja piva zahtijeva visoku razinu automatizacije i strogo kontrolirane procese kako bi se osigurala konzistentna kvaliteta, sigurnost i učinkovitost proizvodnje. Jedan od ključnih aspekata u pivarskoj industriji je higijena proizvodnog sustava, budući da i najmanja kontaminacija može dovesti do neželjenih promjena u okusu, stabilnosti i sigurnosti konačnog proizvoda. Kako bi se osigurali visoki higijenski standardi i povećala učinkovitost procesa, sve se češće implementiraju automatizirani Clean-in-Place (CIP) sustavi.

CIP sustav omogućuje automatsko čišćenje i dezinfekciju unutrašnjosti proizvodnih linija, spremnika i cjevovoda bez potrebe za njihovim rastavljanjem. Proces se provodi pomoću precizno kontroliranih sekvenci ispiranja, čišćenja i dezinfekcije, koristeći različite kemijske otopine, temperaturu i tlak. Automatizacija CIP stanice ne samo da poboljšava higijenske standarde, već i smanjuje potrošnju vode, kemikalija i energije, što doprinosi ekonomskoj i ekološkoj održivosti proizvodnje.

Osim automatizacije procesa čišćenja, dodatno poboljšanje sustava postiže se prikupljanjem podataka sa senzora i aktuatora koji nadziru ključne parametre kao što su temperatura, tlak, protok i koncentracija kemikalija. Ovi podaci se pohranjuju u bazu podataka, gdje se kasnije mogu analizirati kako bi se optimizirale radne karakteristike sustava, detektirali mogući problemi te omogućilo prediktivno održavanje. Obrada i analiza podataka također omogućuju optimizaciju resursa, smanjenje troškova te osiguranje visoke razine ponovljivosti procesa.

U ovom radu analizirat će se principi rada automatiziranog CIP sustava, tehnologije senzora i aktuatora koji omogućuju njegovo precizno upravljanje te metode prikupljanja, obrade i analize podataka u svrhu poboljšanja učinkovitosti i održivosti procesa.



## 2. Brewmaxx sustav

Brewmaxx je napredni sustav automatizacije dizajniran posebno za pivovare i industrije s procesima tekućeg upravljanja. Njegova glavna prednost leži u modularnoj i skalabilnoj arhitekturi koja omogućuje prilagodbu različitim veličinama proizvodnih pogona, od malih zanatskih pivovara do velikih industrijskih postrojenja.[1]

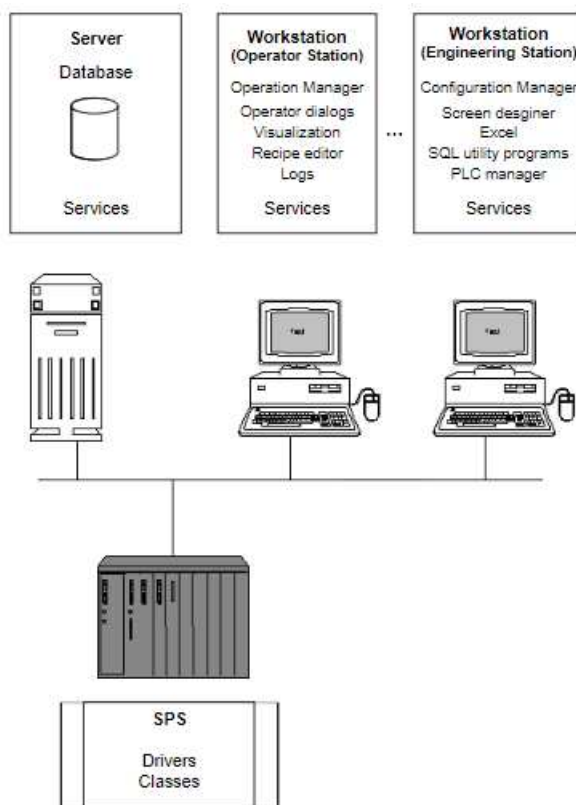
Ovaj sustav omogućuje potpunu kontrolu nad procesima proizvodnje piva, uključujući kuhanje sladovine, fermentaciju, filtraciju i pakiranje. Koristeći napredne algoritme i sučelje prilagođeno korisniku, Brewmaxx optimizira rad strojeva, smanjuje potrošnju resursa te poboljšava dosljednost i kvalitetu krajnjeg proizvoda.

Integracija s različitim upravljačkim sustavima i sensorima omogućuje kontinuirano praćenje ključnih parametara poput temperature, tlaka i protoka, dok sofisticirani sustav alarmiranja i izvještavanja osigurava pravovremeno reagiranje na moguće probleme[2]

Zahvaljujući fleksibilnosti i mogućnosti nadogradnje, Brewmaxx se ističe kao jedno od najpouzdanijih rješenja za pivovare koje žele unaprijediti svoju proizvodnju uz minimalne troškove i maksimalnu učinkovitost.[3]

## 2.1. Brewmaxx arhitektura

Na slici 2.1. prikazana je arhitektura Brewmaxx sustava. Njegova arhitektura sastoji se od nekoliko ključnih komponenti koje su međusobno povezane putem iste mreže radi osiguravanja integriranog nadzora i upravljanja.



Slika 2.1. Brewmaxx arhitektura

### 2.1.1. Server

Server je središnji dio sustava na kojem se nalazi baza podataka i ključne usluge potrebne za rad cijelog sustava. Na serveru se vrtila SQL baza podataka koja pohranjuje sve konfiguracijske podatke.

Aplikacija Configuration Manager, koja je dio sustava, omogućuje definiranje svih komponenti i parametara. Kada se konfiguracija završi, podaci se spremaju na server, čime se osigurava centralizirano upravljanje i spremanje svih informacija.

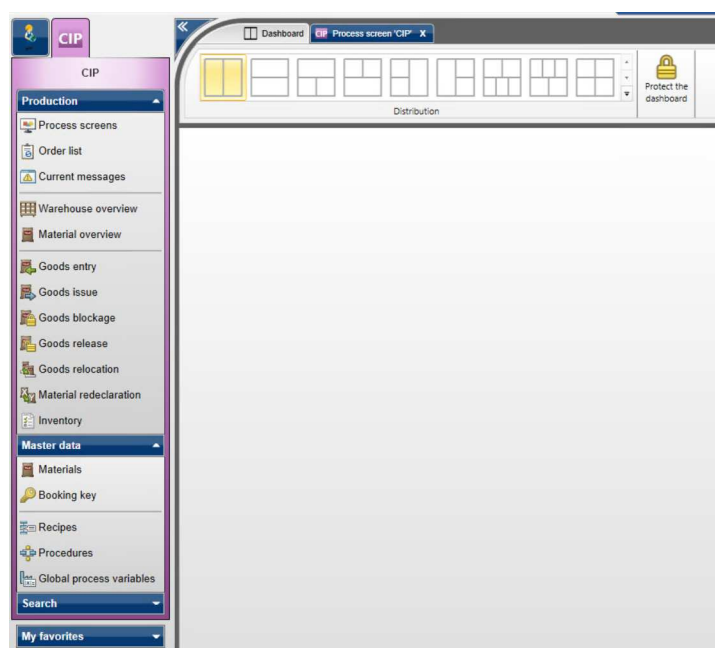
## 2.1.2. Operaterska stanica

Na ovoj radnoj stanici rade operateri koji su odgovorni za praćenje i kontrolu proizvodnih procesa. Njihovo korisničko sučelje uključuje aplikacije poput:

- Operation Manager
- Operator dialogs
- Visualization
- Recipe editor
- Logs

### Operation manager

Operation Manager u sustavu Brewmaxx ključna je aplikacija koja omogućuje operaterima praćenje i upravljanje procesima proizvodnje u stvarnom vremenu. Kroz intuitivno korisničko sučelje, operateri mogu nadzirati parametre kao što su temperatura, tlak i protok, pregledavati alarmne poruke, uređivati recepte te upravljati različitim fazama proizvodnje. Ova aplikacija osigurava brze reakcije na promjene u procesu i optimizira rad sustava radi postizanja maksimalne učinkovitosti i sigurnosti.[4]



Slika 2.2. Početna stranica Operation Managera

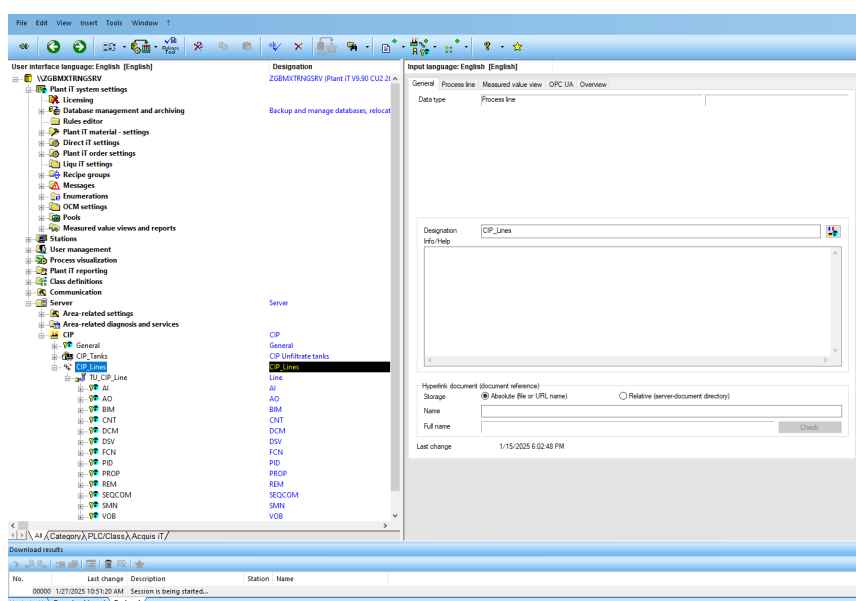
## 2.1.3. Inženjerska stanica

Ova radna stanica namijenjena je inženjerima koji konfiguriraju sustav i razvijaju aplikacije unutar njega. Aplikacije koje se koriste na ovoj stanici uključuju:

- Configuration Manager – konfiguracija svih sustavnih komponenti
- Screen Designer – dizajniranje ekrana i vizualizacijskih prikaza
- PLC Manager – upravljanje komunikacijom s PLC-om
- SQL Utility Programs – rad s bazama podataka

### Configuration Manager

Osnovna funkcija Configuration Managera 2.3. je centralizirano upravljanje postavkama koje se koriste za rad sustava, uključujući definiranje proizvodnih procesa, postavljanje senzora i aktuatora te upravljanje PLC uređajima. Inženjeri koriste Configuration Manager za unos i ažuriranje podataka o svim relevantnim komponentama u sustavu. Ti podaci obuhvaćaju parametre poput radnih temperatura, brzine protoka, električne provodljivosti, tlaka... Konfiguracija se sprema na centralni server, što omogućuje jednostavno ažuriranje i distribuciju postavki svim povezanim uređajima i radnim stanicama. Povezan je s alatima za vizualizaciju i praćenje procesa kako bi operateri i inženjeri uvijek imali uvid u trenutni status sustava.[5]



Slika 2.3. Configuration manager

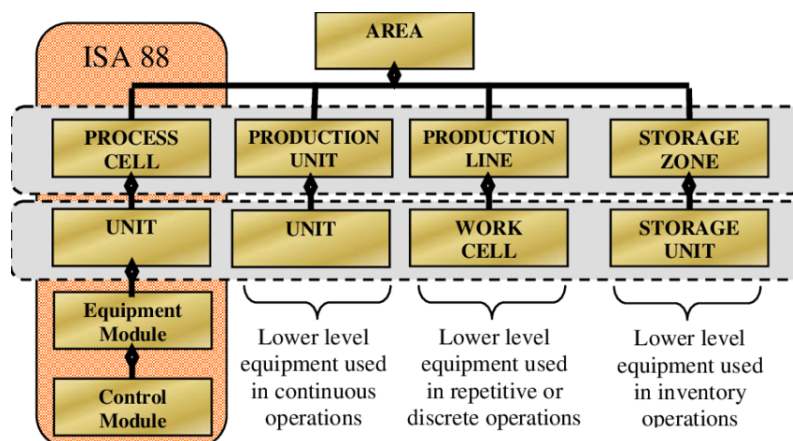
## 2.1.4. Programirajući logički kontroler (PLC)

PLC je programirajući logički kontroler koji upravlja stvarnim procesima u proizvodnji. Nalazi se na istoj mreži kao i ostale komponente sustava. Njegova glavna funkcija je izravna interakcija sa sensorima i aktuatorima u pogonu te izvršavanje upravljačkih instrukcija u stvarnom vremenu.[6]

PLC koristi drajvere i klase koji omogućuju komunikaciju s drugim komponentama sustava i osigurava rad u skladu s konfiguriranim parametrima. Nakon što inženjer izvrši konfiguraciju, podaci se preuzimaju sa servera u PLC, omogućujući kontroleru da upravlja procesima prema zadanim postavkama.

## 2.2. ISA-88: Standard za šaržnu proizvodnju

ISA-88 2.4., poznat i kao standard za šaržnu proizvodnju, međunarodno je priznati okvir koji definira metodologiju upravljanja procesima šaržne proizvodnje u industrijama poput prehrambene, farmaceutske i kemijske. Međunarodno društvo za automatizaciju (ISA) razvilo je ovaj standard kako bi osiguralo dosljednu terminologiju, strukturiranje proizvodnih procesa i definiranje recepata. Omogućuje razdvajanje fizičkih i proceduralnih elemenata, čime pruža fleksibilnost u prilagodbi proizvodnim zahtjevima. Svaki recept unutar šaržne proizvodnje sadrži sve potrebne korake i parametre, dok sustav automatizacije bilježi i prati sve faze kako bi se osigurala dosljednost i sljedivost. Primjenom kontrolnih točaka u procesu poboljšava se kvaliteta proizvoda, a smanjuje se rizik od odstupanja.[7]



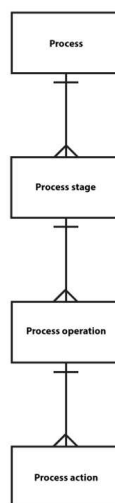
Slika 2.4. Fizički model ISA88 sustava

Automatizacijski sustavi, poput Brewmaxx-a, podržavaju ovaj standard, omogućujući centralizirano praćenje i kontrolu. ISA-88 također omogućuje bolje razumijevanje korisničkih zahtjeva i olakšava integraciju među različitim dobavljačima sustava. Implementacija ovog standarda pomaže tvrtkama povećati učinkovitost, smanjiti troškove te uskladiti se s regulatornim zahtjevima i industrijskim normama.

### 2.3. Proceduralno upravljanje sustavom

Proceduralno upravljanje odnosi se na način upravljanja procesima koji se temelji na definiranju i izvođenju unaprijed određenih koraka ili procedura. Ovaj pristup se često koristi u složenim procesima, poput onih u prehrambenoj, kemijskoj ili farmaceutskoj industriji, gdje je potrebno dosljedno izvesti niz operacija kako bi se postigao željeni rezultat. Proceduralno upravljanje definira što se i kada treba dogoditi tijekom procesa, a upravljački sustav izvršava te korake prema utvrđenom redosljedu.[7] U okviru standarda ISA-88, proceduralno upravljanje se strukturira kroz nekoliko razina:

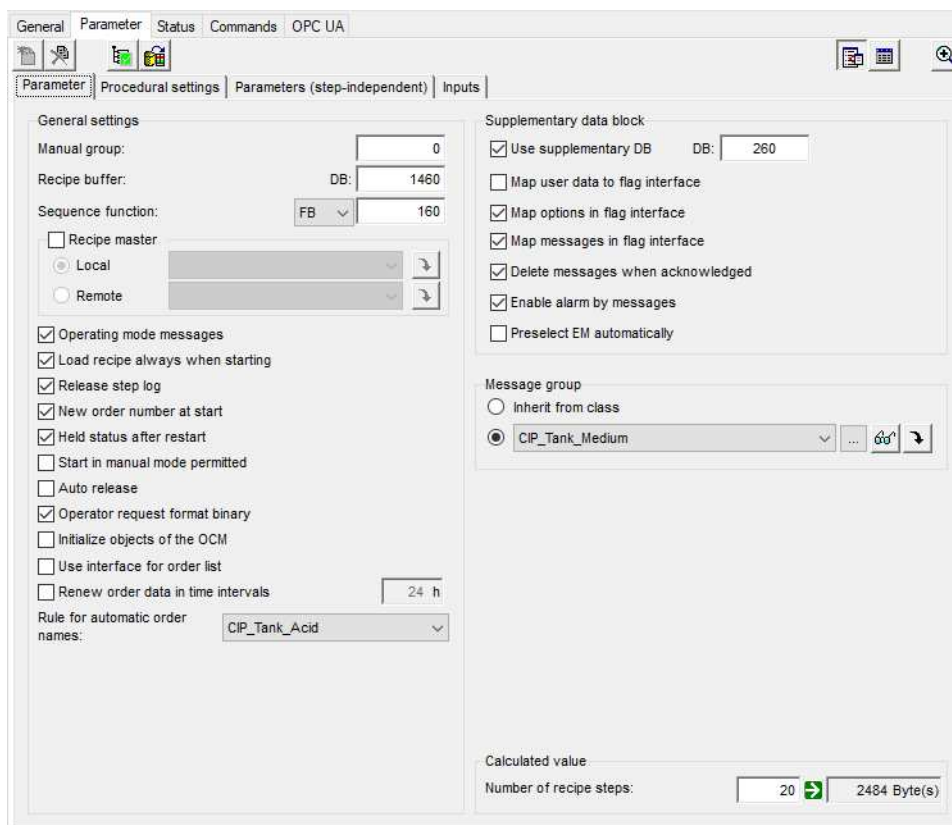
- procedure – glavni tijek operacija unutar procesa
- potprocedure – podijeljeni dijelovi unutar glavne procedure
- operacije – konkretni koraci unutar potprocedure
- faze – najniža razina, gdje se obavljaju stvarne funkcije, poput otvaranja ventila, uključivanja pumpi ili mjerenja parametara



Slika 2.5. Prikaz proceduralnog upravljanja

## 2.3.1. Sekvenca

Sekvenca u Brewmaxxu predstavlja programski modul koji uključuje sve potrebne parametre za izvršavanje određenih operativnih koraka unutar procesne jedinice. Na slici 2.6. prikazan je dio sučelja za konfiguraciju sekvence u procesnom upravljačkom sustavu. Ovdje se definiraju ključne postavke za rad procesne jedinice, uključujući redoslijed koraka, parametre recepta i način upravljanja porukama. Sekvenca prolazi kroz različita



Slika 2.6. Konfiguracija sekvence

stanja, a neka od njih su:

- idle – početno stanje kada sekvenca nije aktivna
- starting – prijelazno stanje pri pokretanju sekvence
- running – sekvenca aktivno izvršava zadane korake
- completed – završno stanje nakon uspješnog izvršenja svih koraka
- prekidna stanja – kada dolazi do neočekivanog prekida ili greške tijekom rada

### **2.3.2. Procesne operacije**

Procesne operacije predstavljaju korake unutar sekvence jedinične procedure u proizvodnom procesu. Svaka procesna operacija sadrži definirane uvjete za prijelaz iz jednog koraka u sljedeći. Sekvenca te korake izvršava točno onim redoslijedom koji je propisan procedurom. Da bi se olakšalo upravljanje i razvoj programske podrške, svaki korak može koristiti predefinirane signale, kao što su StepBegin, StepBeginDelayed, StepPre-End i StepEnd. Ovi signali omogućuju jasnu kontrolu nad početkom, trajanjem i završetkom svakog koraka.

### **2.3.3. Parametri procesa**

Parametri u Brewmaxx-u ključni su za upravljanje procesnim jedinicama i proizvodnim sekvencama. Oni su podijeljeni na nekoliko vrsta, uključujući parametre koraka, procedure, recepta i opreme, od kojih svaki ima specifičnu funkciju i svrhu.

Parametri koraka definiraju rad pojedinih koraka u sekvenci. Ovi parametri obuhvaćaju funkcije poput brojača ili analitičkih vrijednosti (npr. tlak ili temperatura), kao i setpointe, tj. ciljane vrijednosti koje se prate putem senzora. Setpointi osiguravaju da je proces unutar definiranih granica prije prijelaza na sljedeći korak.

Parametri procedure razlikuju se ovisno o proizvodnoj proceduri koja se provodi. Njihova uloga je prilagodba procesa specifičnim zahtjevima proizvodnje, što omogućuje fleksibilnost u različitim vrstama proizvodnih serija.

Parametri recepta definiraju ključne operative postavke za pojedinu seriju, uključujući sastojke, trajanje operacija i druge važne uvjete. Oni su važni za standardizaciju i osiguravanje konzistentnosti krajnjeg proizvoda.

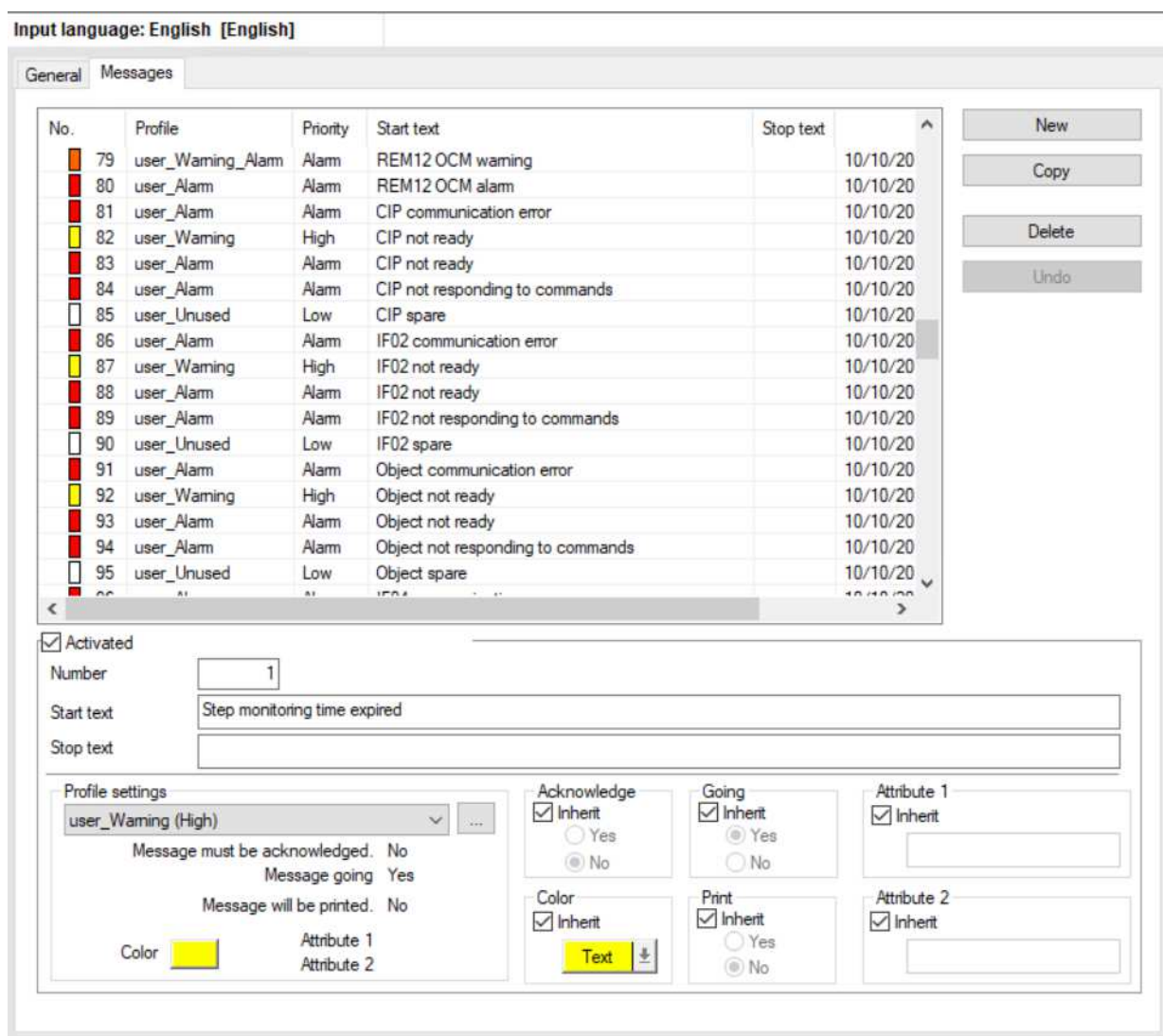
Parametri opreme odnose se na fizička svojstva procesne opreme, poput kapaciteta ili tehničkih ograničenja. Za razliku od drugih parametara, oni su konstantni i ne mijenjaju se tijekom izvršavanja procedura.

Dodatno, postoje i binarni parametri tipa "Option", koji se koriste za omogućavanje ili onemogućavanje određenih opcija u procesu. Ovi parametri služe za jednostavno uključivanje dodatnih funkcija u proizvodnom procesu, s vrijednostima 0 ili 1.



## 2.3.4. Poruke

Poruke u proizvodnom sustavu spremaju se u bazu podataka i razlikuju se prema tekstu, prioritetu, boji i tipu. Ove poruke služe za obavještanje operatera o statusima, greškama ili drugim važnim informacijama u procesu te imaju ključnu ulogu u upravljanju procesima. Ovisno o tipu poruke, put njezinog prikazivanja može varirati. Poruke se prikazuju na HMI sučelju (Human-Machine Interface), unutar Operation Managera, gdje ih operater može pregledati, potvrditi.



Slika 2.7. Prikaz poruka iz Configuration Managera

## 2.4. Upravljačke komponente sustava

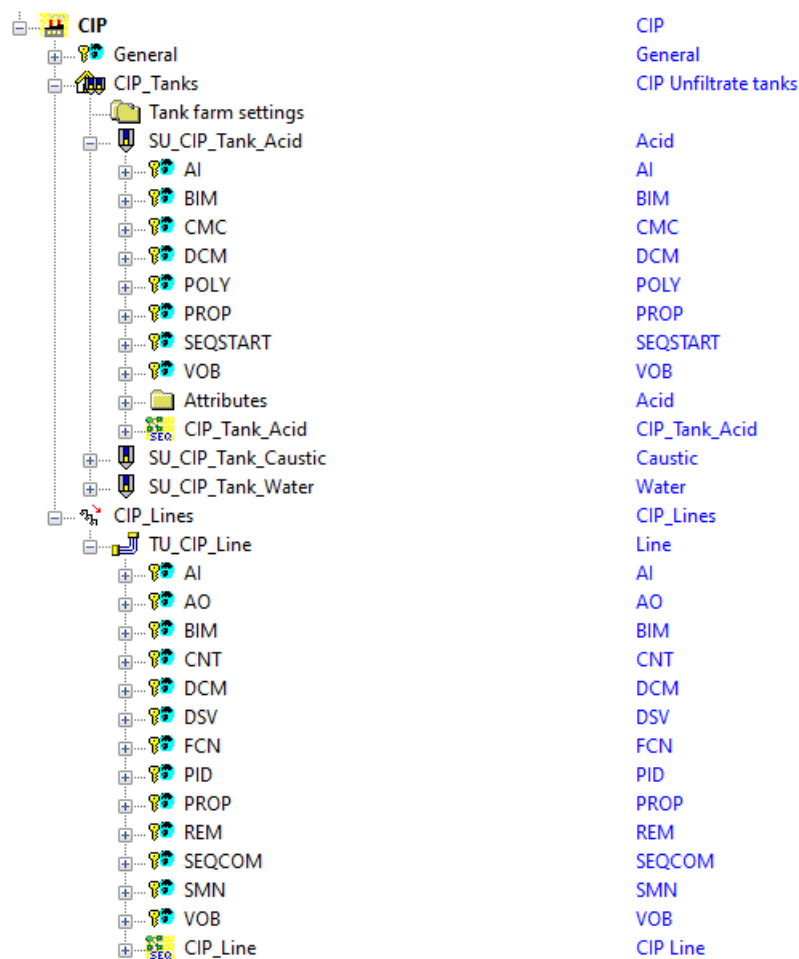
Upravljačka komponenta (Control Module, CM) predstavlja najmanju jedinicu upravljačkog sustava. Može biti programski element, poput matematičkih ili logičkih modula, ili fizička komponenta, poput mjernih instrumenata, ventila, pumpi i motora. Svaka upravljiva komponenta ima svoju specifičnu funkciju u procesu automatizacije.

CM lista je popis svih modula potrebnih za automatizaciju određenog dijela pivovare ili drugog proizvodnog sustava. Implementacijom tih modula omogućuje se integracija u PLC program kroz generiranje podatkovnih blokova (Data Blocks, DB), koji sadrže informacije o stanju i parametrima upravljačkih komponenti. Ova struktura omogućuje efikasan nadzor i upravljanje svim procesima unutar sustava automatizacije.

U procesu su korištene različite upravljačke komponente 2.8. koje omogućuju nadzor, regulaciju i izvršenje funkcija unutar sustava. Ovdje su opisane glavne komponente:

- Class 25 DCM (Direct Control Modules) - aktuatori, mehaničke komponente koje fizički izvršavaju naredbe sustava, primjerice otvaranjem i zatvaranjem ventila ili pokretanjem pumpi. Povezani su s PLC-om za automatsko upravljanje procesom.
- Class 26 DSV (Double Seat Valve) - ventil s dvostrukim sjedištem, posebna vrsta ventila dizajnirana za odvajanje dviju različitih tekućina unutar procesa, bez mogućnosti njihovog miješanja.
- Class 72 BIM (Binary Input with Monitoring) - binarni ulaz s nadzorom, koristi se za prijem digitalnih signala (npr. stanje uključen/isključen) s prekidača, sigurnosnih sklopki ili drugih uređaja. Uključuje funkcije nadzora za detekciju kvarova ili neispravnosti. Također služi za detekciju napunjenosti tanka, prateći tri ključna stanja: prazan, pun i nominalna razina. Ova funkcionalnost omogućuje sustavu preciznu regulaciju operacija poput punjenja, pražnjenja i održavanja optimalne razine tekućine.
- Class 87 FCN (Frequency Converter) - frekvencijski pretvarač, ova komponenta upravlja brzinom rotacije motora reguliranjem frekvencije napajanja, tako omogućuje precizno upravljanje brzinom protoka tekućine ili zraka u procesu.

- Class 88 AI (Analog Input) - analogni ulaz, prima analogne podatke s različitih senzora, poput onih koji mjere temperaturu, tlak, razinu tekućine ili provodljivost. Ovi senzori šalju kontinuirane signale koji predstavljaju fizičke veličine. Analogni ulazi unutar sustava automatizacije rade kao A/D pretvornici, pretvarajući te analogne signale u digitalne vrijednosti koje može obraditi upravljački sustav.
- Class 89 AO (Analog Output) - analogni izlaz, koristi se za upravljanje uređajima koji zahtijevaju kontinuiranu regulaciju, poput regulacijskih ventila i pumpi s frekvencijskim pretvaračem. Funkcionira kao D/A pretvornik (digitalno-analogni pretvornik), pretvarajući digitalne naredbe iz upravljačkog sustava u analogne signale.
- Class 97 CMC (Extended Drive Control Modules) - prošireni moduli za upravljanje pogonima, omogućuju napredno upravljanje pogonima, poput pumpi ili motora, pružajući dodatne funkcije kao što su nadzor rada, regulacija brzine i zaštita od preopterećenja.



**Slika 2.8.** Prikaz upravljačkih komponenata procesa

## 2.4.1. Upravljanje upravljačkim komponentama

Nakon povezivanja fizičkih komponenti s odgovarajućim upravljačkim modulima, potrebno je omogućiti njihovu aktivaciju i nadzor. Umjesto da se ove operacije obavljaju direktno u kodu, koristi se upravljačka matrica (engl. Object Control Matrix) 2.10. Ovaj pristup donosi brojne prednosti, posebice kod složenih industrijskih sustava, jer omogućuje centralizirano upravljanje objektima, uključujući i one povezane s različitim PLC uređajima. Tako se pojednostavljuje nadzor i koordinacija velikog broja procesa.

Struktura upravljačke matrice organizirana je tako da stupci predstavljaju aktivnosti koje se mogu izvršavati nad određenim komponentama. Te aktivnosti potrebno je unaprijed definirati i dodati u aktivacijsku listu (engl. Activity listu) 2.9., koja se koristi za upravljanje procesima. Svakoj aktivnosti u matrici dodjeljuje se razina greške koja može nastati tijekom njezina izvršavanja. Ove razine grešaka kategorizirane su tako da P1 označava upozorenja, dok P3 predstavlja alarme koji zahtijevaju zaustavljanje sekvence. Takva kategorizacija omogućuje generiranje poruka prema skupinama aktivnosti, čime se znatno ubrzava proces programiranja jer nije potrebno ručno kodirati poruke za svaku pojedinu aktivnost.

The screenshot shows a software configuration window with tabs for 'General', 'Activities', and 'Use'. The 'Activities' tab is active, displaying settings for error handling and a table of activities.

**Error handling settings:**

- Status check during reservation: Bitmask must be set: 0x00000000, Bitmask must not be set: 0x00000000
- Ignore bitmask (!): 0x00000002
- Bitmask severity level 1 (P1): 0x00000005, Bitmask severity level 2 (P2): 0x00000000
- Bitmask severity level 3 (P3): 0x487417F8, Bitmask severity level 4 (P4): 0x00000000

**Activity List Table:**

No	Name	Designation	Tolerance	Error handling					Accept status		Info/Help for activity
				I	P1	P2	P3	P4	Set mask	Reset mask	
0	Idle	Idle	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
1	InherentlySafe	InherentlySafe	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
2	OCM warning	OCM warning	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
3	OCM alarm	OCM alarm	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
4	Bypass	Bypass	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
5	Drain switch	Drain switch	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
6	Drain	Drain	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
7	Pump fix	Pump fix	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
8	Pump control	Pump control	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
9	Heating fix	Heating fix	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
10	Heating control	Heating control	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
11	Tank805 supply	Tank805 supply	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
12	Tank806 supply	Tank805 supply	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
13				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
14				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
15				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
16				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
17				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
18	FWT supply	FWT supply	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0x00000000	0x00000000	
19				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Slika 2.9. Aktivacijska lista

Redovi u upravljačkoj matrici predstavljaju upravljačke module povezane s fizičkim komponentama. Na sjecištima redova i stupaca definiraju se specifične naredbe koje određuju kako će se komponenta upravljati. Primjerice, za ventil se koriste naredbe kao što su "C" za aktivaciju, "N" za provjeru otvorenosti, "F" za provjeru zatvorenosti te "E" za provjeru greške. Slično tome, za upravljanje motorima koriste se naredbe "CN" za vrtnju u smjeru kazaljke na satu ili sporu vrtnju, te "CF" za vrtnju u suprotnom smjeru ili brzu vrtnju.

Aktivacija komponente u PLC-u provodi se postavljanjem odgovarajućeg bita, koji ima redni broj definiran u aktivacijskoj listi. Nakon uspješne aktivacije postavlja se nadzorni bit (OCM\_Feedback), koji pruža povratnu informaciju o statusu komponente. Taj nadzorni bit koristi se dalje u kodu kao uvjet za izvođenje drugih aktivnosti. Ovakav sustav omogućuje pouzdano i koordinirano izvršavanje svih operacija, a njegova struktura povećava fleksibilnost i preglednost prilikom upravljanja složenim procesima. Na ovaj način omogućuje se brža implementacija, bolji nadzor nad sustavom i smanjenje mogućnosti pogrešaka u radu postrojenja.

Input language: English [English]

General Structure Use

Matrix: CIP\_Line  
CIP\_Line

Activity list: al\_52\_063\_CIP\_L3

Bitmask must be set: 0x00000000  
Bitmask must not be set: 0x00000000

Select activity list  
Embed OCM  
Add position  
Add objects

Individual objects

WZGBMXTRI  
Server  
CIP

Object name	Class	Station	Designation	0	1	2	3	4	5	6	7	8	9	10	11	12	18	20	21	22
Idle	Inherer	OCM	al																	
Idle	Inherer	OCM	al																	
Tolerance:	0	0	5	5	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
Set mask:	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
Reset mask:	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
Objects																				
V_CL_B	25	1	Bypass valve	FE			E	C												
V_CL_D	25	1	Drain valve	FE			E				C									
FC_CL_S	87	1	Supply pump drive	FE			E													
PID_FT_CL_S	91	1	Supply flow controller	FE			E					CN	CN							
V_CL_S_CVR	25	1	Steam supply control valve release	FE			E							C	C					
PID_CL_TT_S	91	1	Supply temperature controller	E			E						N	C						
V_CL_CTA_S	25	1	Acid supply valve	FE			E									C				
V_CL_CTC_S	25	1	Caustic supply valve	FE			E										C			
V_CL_CTW_S	25	1	Fresh water supply valve	FE			E											C		
V_CL_HW_S	25	1	Hot water supply valve	FE			E												C	
V_CL_CTA_R	25	1	Acid return valve	FE			E													C
V_CL_CTC_R	25	1	Caustic return valve	FE			E													
V_CL_S_S	25	1	Steam supply valve	FE			E													C
CT_CL_R	73	1	Return conductivity transmitter	E			E													
FS_CL_R	72	1	Return flow switch	E			E													
V_CL_C	26	1	Circulation valve	FE			E													

Legend

C Activate S Ac  
N Monitoring ON P Ac  
F Monitoring OFF V Vo  
E Monitoring Failure

Information

Matrix size in bytes: 700 Matrix number: 6663  
Number of activities: 19 Active objects: 16 Pipes: 0  
Number of objects: 16 Passive objects: 0 Pipes with status: 0  
Number of stations: 1 CA's: 0 Other objects: 0

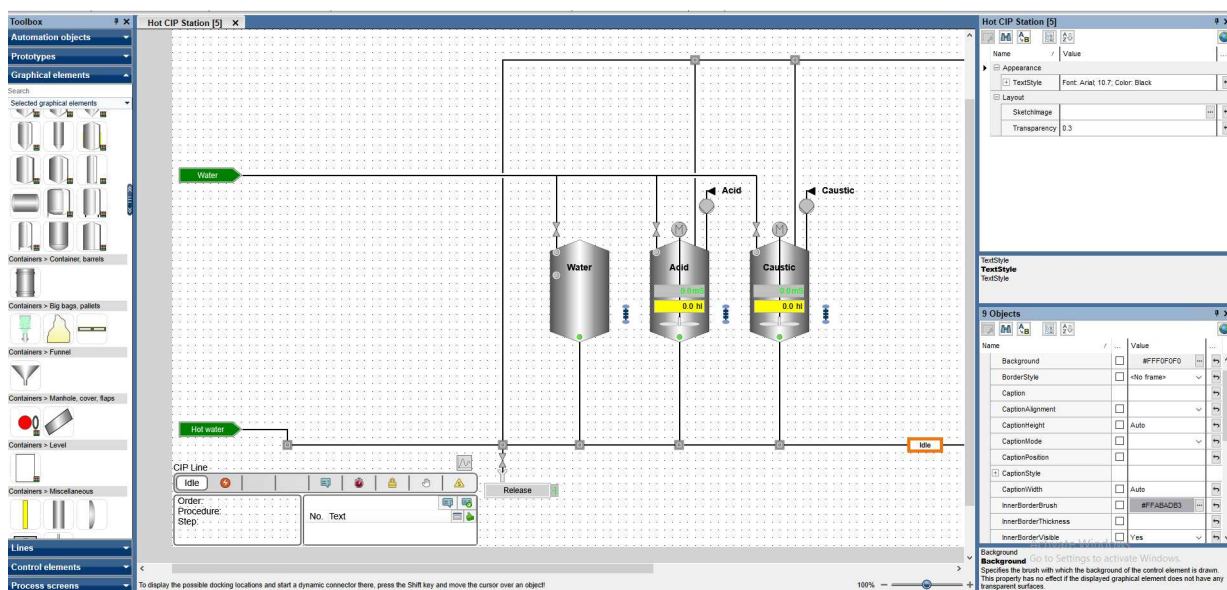
Slika 2.10. Upravljačka matrica

## 2.5. Vizualizacijsko sučelje

SCADA (Supervisory Control and Data Acquisition) sustav predstavlja ključnu tehnologiju za nadzor, prikupljanje podataka i upravljanje industrijskim procesima. Omogućuje operaterima pregled stanja procesnih jedinica, prikaz mjernih podataka i upravljanje opremom u stvarnom vremenu. U slučaju grešaka ili kritičnih događaja, operaterima se automatski prikazuju poruke s odgovarajućim upozorenjima, čime se omogućuje brza reakcija i osigurava sigurnost sustava.

SCADA sustav obuhvaća HMI (Human-Machine Interface), koji prikazuje informacije kroz različite grafičke elemente poput dijagrama, pokazivača, tekstualnih poruka i vizualnih prikaza. Podaci u sustav dolaze iz PLC-a (Programabilnog logičkog kontrolera), a dizajn aplikacije često prati izgled dijagrama cjevovoda i instrumentacije (engl. Piping and Instrumentation Diagram – P&ID) sheme, kako bi se olakšalo snalaženje operaterima, jer takav prikaz odražava stvarni raspored opreme u postrojenju.

U Brewmaxx-u, SCADA aplikacija se realizira pomoću alata Process Screen Designer 2.11., koji je dio konfiguratora. U aplikaciji se koriste različiti grafički elementi, uključujući linije, vizualizacijske prikaze, tekstualne okvire i poveznice na druge ekrane. Na tim povezanim ekranima detaljnije su prikazani pojedini dijelovi postrojenja, čime se omogućuje pregled svih dijelova procesa s različitih lokacija unutar sučelja.



Slika 2.11. Process Screen Designer

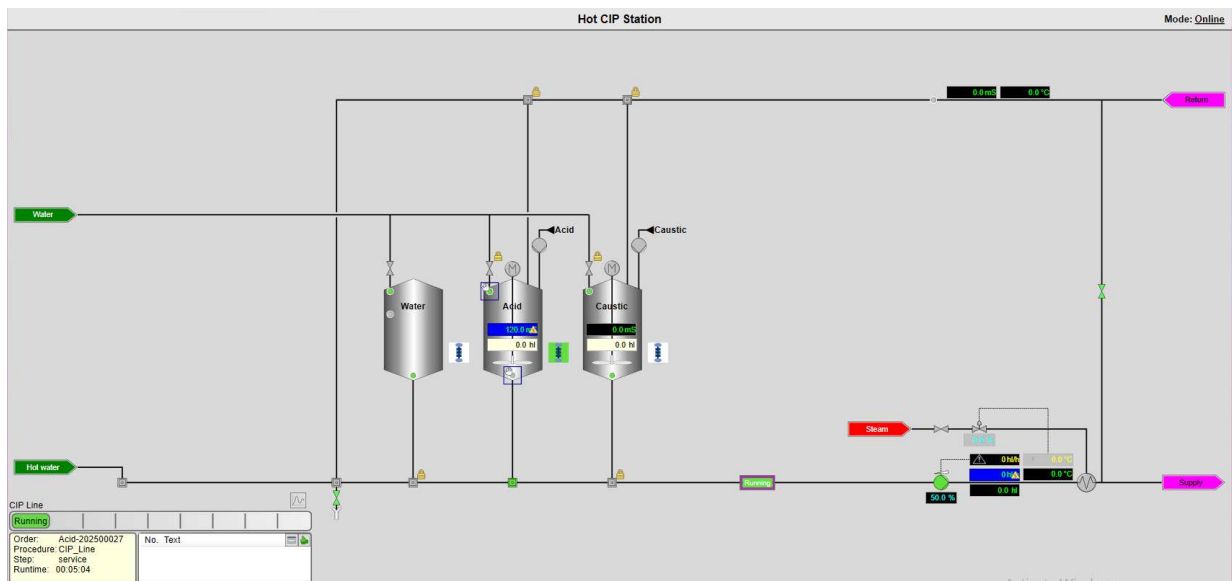
### **3. Automatizacija sustava za pranje objekata**

U industrijskim sustavima upravljački program se aktivira pokretanjem određene sekvence iz HMI (Human Machine Interface) sučelja. Otvaranjem sekvence omogućava se učitavanje recepture, a nakon odabira željene recepture, njezin se kod zapisuje u odgovarajući podatkovni blok PLC-a (Programmable Logic Controller). Aktivira se funkcijski blok koji iz baze podataka preuzima sve potrebne parametre i vrijednosti za izvođenje odabranog recepta. Svaka receptura ima definiranu proceduru koja se automatski pokreće, nakon čega započinje izvršenje koraka (PROP).[8]

Automatizacija sustava uključuje razvoj programske podrške, što podrazumijeva izradu upravljačkog programa i vizualizacijskog sučelja procesa. Korisnički zahtjevi u procesnoj industriji definiraju se putem dijagrama cjevovoda i instrumentacije (P&ID) te funkcionalnog opisa rada postrojenja (engl. Functional Design Specification – FDS). Nakon što se postigne dogovor o korisničkim zahtjevima i stekne njihovo potpuno razumijevanje, pristupa se sljedećoj fazi – programiranju i konfiguriranju sustava. U toj fazi potrebno je projektirati SCADA sustav, izraditi PLC program te postaviti i prilagoditi parametre, operacije, aktivacije i ostale funkcije unutar Operation Managera i Configuration Managera.

#### **3.1. Sustav za pranje objekata**

Sustav za pranje objekata (engl. Clean-In-Place – CIP) omogućuje automatsko čišćenje unutrašnjih površina procesne opreme, cijevi, pumpi i ventila bez rastavljanja sustava. Sastoji se od tri tanka – za vodu, kiselinu i lužinu – koji su povezani mrežom ventila, pumpi, senzora i cijevi. Stoga se ovaj sustav sastoji od četiri sekvence, tri sekvence za svaki tank te sekvenca za CIP liniju.



**Slika 3.1.** Sustav za pranje objekata iz Operation Managera

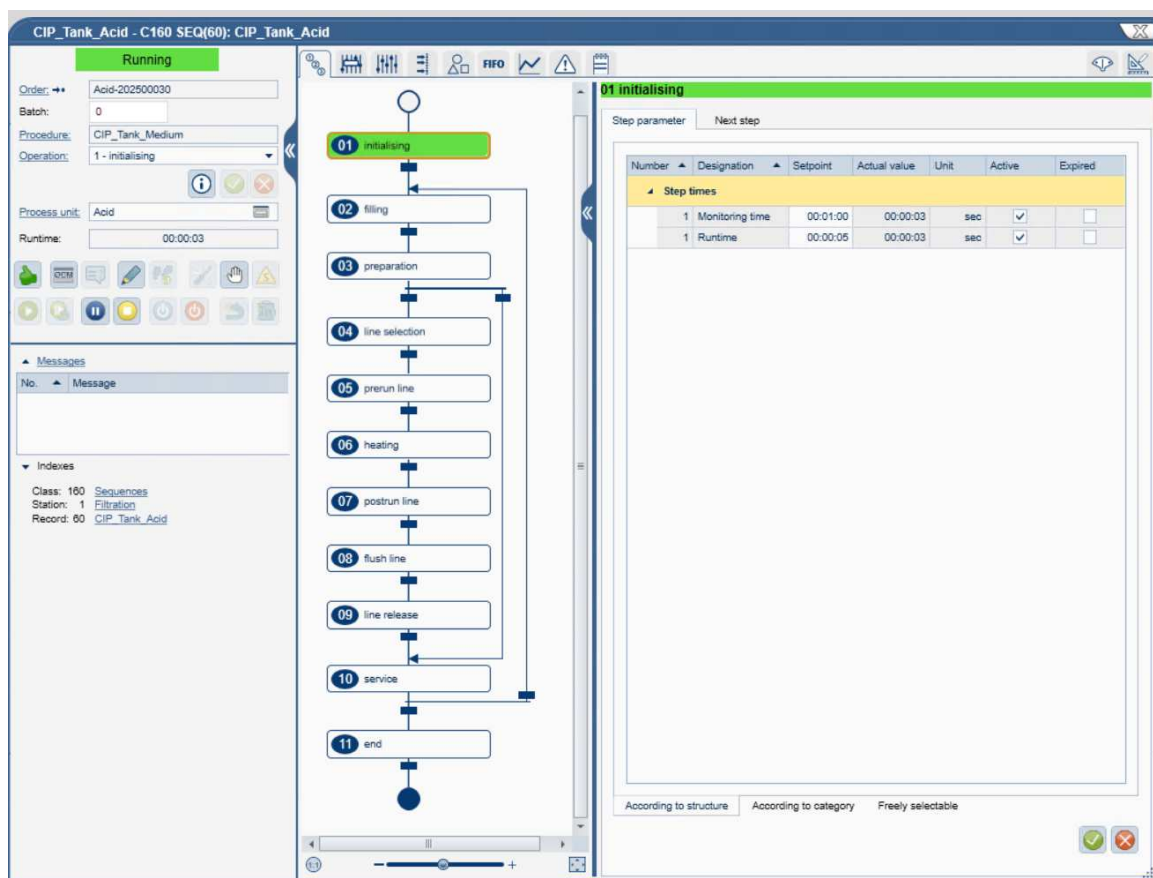
Proces se odvaja u dvije ključne faze: pripremu smjese i distribuciju prema objektima. U prvoj fazi puni se tank za vodu čistom vodom, dok se tankovi za kiselinu i lužinu također najprije pune vodom. U tim tankovima kemikalije se dodaju putem pumpi, a smjesa se miješa pomoću ugrađenih agitatora. Agitatori rade prema pulse-pause metodi, gdje se miješanje izvodi u ciklusima pulsa i stanke, čime se osigurava učinkovito i ravnomjerno spajanje kemikalija s vodom. Kad senzori za provodljivost potvrde postizanje odgovarajuće koncentracije, otvara se ventil na tanku i uključuje pumpa koja pokreće cirkulaciju smjese kroz cijevi. Smjesa prolazi kroz grijač, gdje se pomoću pare i PID regulatora postiže optimalna temperatura. Zagrijana smjesa nastavlja cirkulirati kroz sustav i vraća se u tank.

Nakon pripreme smjese, sustav prelazi u fazu distribucije. Na zahtjev objekta koji se čisti, otvara se ventil na odabranom tanku, a pumpa potiskuje smjesu kroz cjevovod. Protok se regulira mjeračem protoka i PID regulatorom. Smjesa prolazi kroz objekt, čisti ga te se vraća u sustav. Povratni senzori prate parametre smjese poput provodljivosti i temperature. Ako parametri zadovoljavaju, smjesa se vraća u tank, a u suprotnom ide prema odvodu. Proces završava ispiranjem cijelog sustava vodom. Također važno je naglasiti da je objekt (koji se priprema ili koji se čisti) ključni element u sustavu jer on šalje signale sustavu te upravlja procesom čišćenja. Objekt signalizira kada je potrebno aktivirati određene ventile i pumpe u određenim koracima.



## 3.2. Sekvenca tankova

Tankovi za kemikalije u CIP procesu prolaze kroz sve korake sekvence, uključujući inicijalizaciju, punjenje, preparaciju, prerun, zagrijavanje, postrun, ispiranje, oslobađanje linije, servis i kraj, dok tank s vodom ima samo tri koraka: inicijalizacija, servis i kraj.



Slika 3.2. Puni prikaz sekvence tanka

### 3.2.1. Korak 1: Inicijalizacija (Initialising)

Prvi korak u CIP procesu je inicijalizacija, čiji je cilj osigurati ispravnost i spremnost svih komponenata sustava za početak rada. U ovom koraku provjerava se početno stanje ključnih dijelova, uključujući ventile, pumpe i senzore. Svi ventili trebaju biti zatvoreni, a pumpa ugašena te se provjerava da nijedan medij ne prolazi kroz sustav. Inicijalizacija se aktivira pokretanjem nultog koraka u upravljačkoj matrici, ("OCM\_Act00 – Idle"). Ova

funkcija postavlja sustav u stanje mirovanja i osigurava da je sustav spreman za sljedeće korake sekvencijalnog procesa čišćenja. Također u ovom koraku resetiraju se svi brojači id parametri vezani uz proces. Vrijednosti se vraćaju na početnu postavku nula, čime se osigurava da sustav nema zaostalih vrijednosti iz prethodnih ciklusa. Time se sprječavaju potencijalne nepravilnosti u radu ventila, pumpe ili upravljačkog sustava. Za prijelaz u drugi korak sekvence potrebno je ispuniti sljedeće uvjete: OCM (upravljačka matrica) mora biti učitana i spremna, tank mora biti prazan, vrijeme izvršavanja trenutačnog koraka (step runtime) mora isteći. Step runtime služi za kontrolu trajanja svakog pojedinog koraka u sekvencijalnom procesu, osiguravajući da sve radnje unutar koraka budu pravilno izvršene, ali i da proces ne traje duže nego što je potrebno. Njegova glavna uloga je održati kontinuitet i stabilnost u radu sustava, sprječavajući greške uzrokovane prekratkim ili predugim zadržavanjem u pojedinom koraku. Tijekom procesa, step runtime osigurava minimalno trajanje svakog koraka kako bi se sve potrebne operacije, poput otvaranja i zatvaranja ventila, aktivacije pumpi ili miješanja medija, mogle ispravno odvit. Istovremeno, omogućuje sustavu da izbjegne "zastoj" u koraku – ako određeni uvjeti ne budu ispunjeni, istekom definirane vrijednosti step runtimea, sustav izlazi iz tog koraka i može aktivirati mehanizme za rješavanje problema ili prijelaz u sljedeću fazu. Prijelaz je moguć tek kada step runtime istekne i svi drugi uvjeti, kao što su spremnost upravljačke matrice (OCM) ili stanje tanka, budu ispunjeni.

```
//Uvjeti za završetak koraka inicijalizacije
// Step end condition 1 - OCM loaded & ready
A      "DB_LiT_C160_SEQ_IF".SEQ.sOCM_ready
A      "OCM_Feedback00"
AN     "StepEnd"
=      "Option 001"

// Step end condition 2 - Tank empty
AN     "Message 182"
O      #SEQ.UserData.OperAck1
AN     "StepEnd"
=      "Option 002"
```

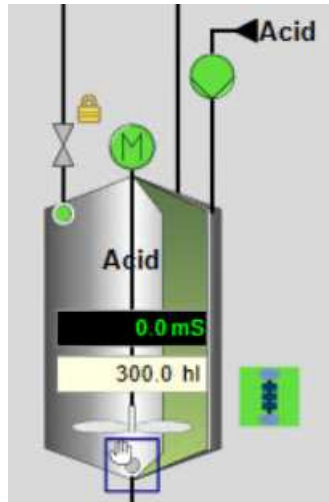
Prvi uvjet provjerava je li OCM (upravljačka matrica) učitana i spreman. Signal sOCM\_ready iz baze podataka (DB\_LiT\_C160\_SEQ\_IF) i povratni signal OCM\_Feedback00 moraju biti aktivni kako bi se potvrdilo da je matrica spremna za rad. Uz to, uvjet zahtijeva da signal StepEnd nije aktivan, što sprječava automatski završetak koraka dok se svi kriteriji ne ispune. Kada su oba signala spremnosti potvrđena, opcija "Option 001" se aktivira, što označava da je prvi uvjet za završetak koraka ispunjen. Drugi uvjet završetka koraka odnosi se na stanje tanka. Kod provjerava negativni signal Message 182, koji označava da je tank prazan. Alternativno, korisnik može ručno potvrditi pražnjenje tanka aktiviranjem signala OperAck1 (potvrda operatera). Ako je jedan od ovih uvjeta ispunjen i signal StepEnd nije aktivan, aktivira se opcija "Option 002", koja označava da je tank prazan i da je drugi uvjet za završetak koraka zadovoljen.

### **3.2.2. Korak 2: Punjenje (Filling)**

Nakon uspješno dovršene inicijalizacije, sustav prelazi u drugi korak sekvence, koji se naziva punjenje. Cilj ovog koraka je napuniti odgovarajući tank vodom do zadane razine kako bi se osigurala dovoljna količina tekućine za nastavak procesa čišćenja. U ovom koraku otvara se ventil za dovod vode, omogućujući ulazak vode u tank. Kako voda ulazi, razina tekućine se kontinuirano prati pomoću senzora za razinu (level sensor) smještenog u tanku. Sustav provjerava očitavanja senzora kako bi se osiguralo postizanje definirane količine vode. Kada senzor registrira da je razina vode dostigla zadanu vrijednost, ventil za vodu se zatvara, čime se završava korak punjenja. Tako se osigurava optimalna količina vode za daljnje faze procesa, poput miješanja kemijskih smjesa, zagrijavanja i distribucije kroz sustav. Nakon punjenja, sustav prelazi u korak preparacije, gdje započinje miješanje i priprema kemijske smjese.

### **3.2.3. Korak 3: Preparacija (Preparation)**

Nakon što je korak punjenja uspješno završen, sustav prelazi u korak preparacije, čija je svrha priprema radne smjese za čišćenje. Ovaj korak uključuje dodavanje kemikalija i miješanje kako bi se postigla odgovarajuća koncentracija i homogena smjesa. U tankovima za kiselinu i lužinu, nakon što se tankovi napune vodom, kemikalije se dodaju putem pumpi. Tijekom ovog procesa, smjesa se miješa pomoću agitatora koji rade prema pulse-pause metodi. Paralelno s miješanjem, senzori za provodljivost kontinuirano prate



**Slika 3.3.** Prikaz završenog punjenja tanka vodom i početka koraka preparacije

koncentraciju smjese u tankovima. Kada provodljivost dostigne zadanu vrijednost, sustav prelazi u idući korak.

```
// Rad agitatora prema pulse-pasuse metodi
// Agitator on time
    A    "OCM_Feedback11"           // Agitator
    AN   "StepEnd"
    =    "VCM_Release02"

    A    "VCM_Result03"           // Agitator off time
    AN   "StepEnd"
    =    "VCM_Trigger02"

// Agitator off time
    A    "VCM_Result02"           // Agitator on time
    AN   "VCM_Result03"          // Agitator off time
    AN   "Held"
    AN   "StepEnd"
    =    "VCM_Release03"

    A    "VCM_Result03"           // Agitator off time
    AN   "StepEnd"
    =    "VCM_Trigger03"
```

U fazi rada agitatora, signal "OCM\_Feedback11" pokazuje da je agitator aktivan. Provjerava se uvijek "StepEnd", koji označava završetak trenutnog koraka. Ako korak nije završen, izlaz "VCM\_Release02" se aktivira i agitator nastavlja raditi. Paralelno se priprema sljedeći korak koji deaktivira agitator. Kada se ispune uvjeti za kraj vremena rada, signal "VCM\_Trigger02" aktivira početak faze mirovanja agitatora. Nakon toga slijedi faza mirovanja agitatora. U ovoj fazi program provjerava nekoliko uvjeta. Prvo, uvjet "VCM\_Result02" potvrđuje da je završeno prethodno aktivno vrijeme agitatora. Zatim se provjerava da agitator nije u stanju rada ("VCM\_Result03"), da stanje pauze ("Held") nije aktivno i da trenutni korak još nije dovršen ("StepEnd"). Kada su svi ovi uvjeti zadovoljeni, postavlja se izlaz "VCM\_Release03", koji označava kraj faze mirovanja agitatora. Istovremeno, signal "VCM\_Trigger03" pokreće povratak u fazu rada agitatora, čime se ciklus ponavlja. Ova metoda osigurava da agitator radi u precizno određenim ciklusima miješanja i stanke, što omogućuje ravnomjerno i učinkovito miješanje kemijske smjese u tankovima.

#### **3.2.4. Korak 4: Odabir linije (Line selection)**

Nakon završene faze preparacije, sustav prelazi u fazu line selection. U ovom koraku, sekvenca linije se pokreće pomoću ugrađene metode definirane kao klasa 170 pod nazivom SeqStart3.5. Ova metoda omogućuje odabir procedure ili recepta koji će se izvršiti, kao i definiranje iz koje sekvence se proces pokreće, koja će sekvenca biti aktivirana te koji bit će poslužiti kao izvor za aktivaciju sekvence (source for activation). Za ovu svrhu koristi se "Option 21" aktivacijski bit. Taj bit aktivira početak rada sekvence linije i osigurava njeno pravilno izvršavanje. Također sustav koristi enumeracije za odabir smjera protoka smjese. Za opskrbljivanje (engl. supply) postoje opcije koje obuhvaćaju dva kemijska tanka, tank za vodu i dovod vruće vode. Za povratak (engl. return), enumeracije omogućuju odabir povratne destinacije – povrat u jedan od dva kemijska tanka, odvod ili cirkulacijski krug. Ovi odabiri definiraju kako će smjesa putovati kroz liniju i na koji način će se vratiti u sustav.

Komunikacija unutar linije i između sekvenci tankova temelji se na FIFO (First In, First Out) metodi. FIFO struktura osigurava redosljednu obradu zahtjeva i naredbi, sprječavajući kašnjenja ili konflikte. Svaki zahtjev koji linija primi obrađuje se prema redosljedu ulaska, čime se omogućuje stabilan i kontroliran rad. Uz to komunikacija

General	
Value	Designation
▶ 0	None
60	Sterile water 805
61	Caustic 806
68	Fresh water
70	Hot water

(a) Enumeracija za opskrblljivanje (supply)

General	
Value	Designation
▶ 0	None
60	Sterile water 805
61	Caustic 806
71	Drain
72	Object circulation

(b) Enumeracija za povratak (return)

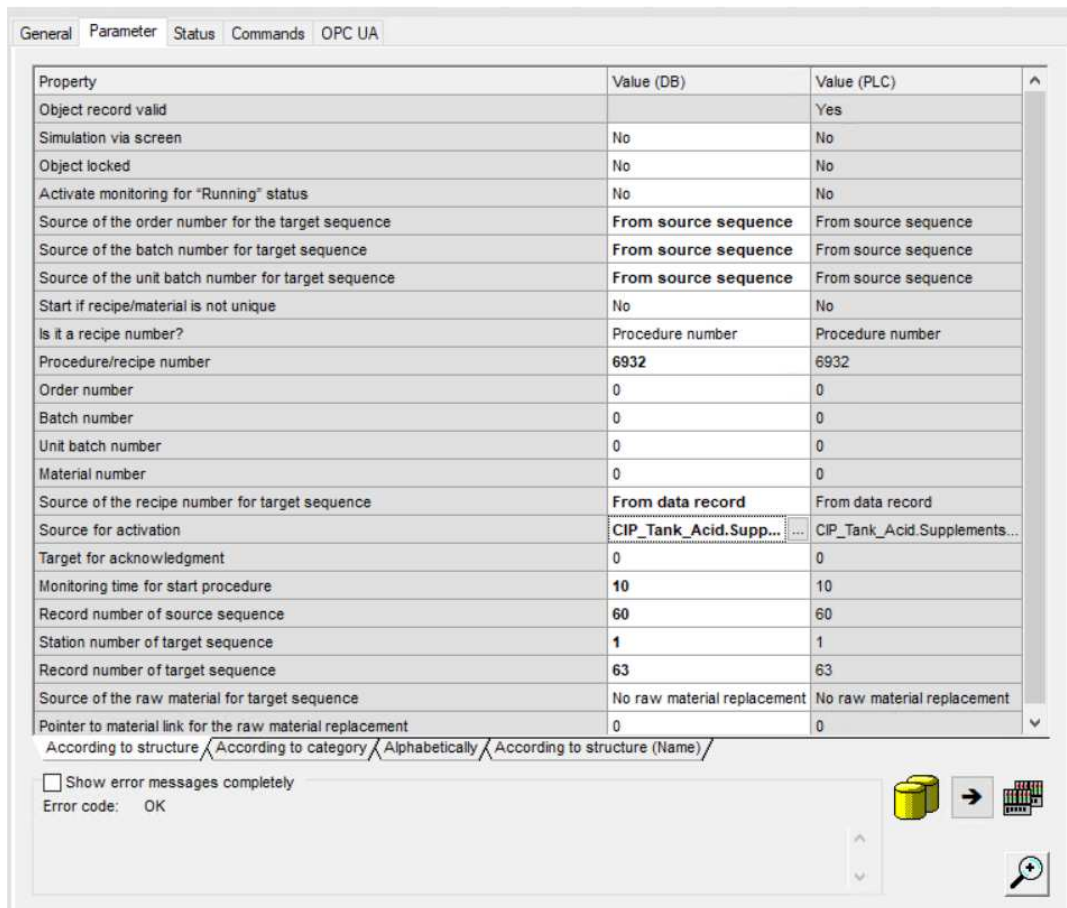
**Slika 3.4.** Prikaz enumeracija za opskrblljivanje (supply) i povratak (return)

između objekta i linije ostvaruje se pomoću klase 169 pod nazivom SeqCom. Ova klasa omogućuje da objekt, liniji šalje upute o tome "što treba napraviti". Budući da se nalazimo u fazi pripreme tanka, tank u ovom kontekstu funkcionira kao objekt koji inicira i upravlja komunikacijom s linijom. SeqCom omogućuje razmjenu informacija između sustava, osiguravajući da linija i tank sinkronizirano surađuju tijekom procesa čišćenja.

Nakon što je sekvenca linije uspješno aktivirana putem metode SeqStart, a komunikacija između tanka i linije uspostavljena putem SeqCom klase, faza odabira linije završava. Sustav je tada spreman za sljedeći korak – distribuciju smjese kroz odabranu liniju prema objektu koji zahtijeva čišćenje.

```
// Pokretanje CIP linije
AN    #SEQ.CIPL3.ComError
A     #SEQ.CIPL3.Selection
A     #SEQ.CIPL3.SelectionOk
A     #SEQ.CIPL3.Fdb01
A     "30SecTact"
AN    "StepEnd"
=     "Option 021"
```

Ovaj dio programa provodi provjeru i aktivaciju opcije "Option 021" u fazi odabira linije (line selection). Prvo se provjerava komunikacija između sustava i linije putem uvjeta #SEQ.CIPL3.ComError. Ako ovaj uvjet nije zadovoljen, odnosno ako nema komunikacijskih grešaka, sustav nastavlja s daljnjim provjerama. Sljedeći uvjet #SEQ.CIPL3.Selection osigurava da je linija odabrana, dok uvjet #SEQ.CIPL3.SelectionOk potvrđuje da je taj izbor validiran i ispravan. Istovremeno, povratna informacija #SEQ.CIPL3.Fdb01 osigu-



**Slika 3.5.** Konfiguracija SeqStart-a

rava da je linija spremna za rad. Uz ove tehničke provjere, vremenski uvjet "30SecTact" osigurava da je prošlo 30 sekundi stabilizacije sustava prije nego što se omogući daljnje izvršenje. Ako su svi uvjeti ispunjeni i korak nije završen, logika aktivira "Option 021" tj. aktivacijski bit za pokretanje sekvence linije.

```
//Tank omogućuje FIFO komunikaciju
AN    #SEQ.SU.IF[6].SU.Selection_ok
AN    "StepEnd"
=     #SEQ.SU.sPrgRelease[7]      // CIP L3 supply

AN    "StepEnd"
=     #SEQ.SU.sPrgRelease[15]    // CIP L3 return
```

Ovdje je prikazana logika koja definira komunikaciju putem FIFO metode između različitih dijelova sustava tijekom odabira linije i aktivacije sekvence. Ključni uvjeti i operacije uključuju provjere statusa i postavljanje izlaza koji kontroliraju protok sekvenci. U

prvom dijelu koda provjerava se uvjet #SEQ.SU.IF[6].SU.Selection\_ok, koji označava je li izbor linije uspješno potvrđen. Ako uvjet nije zadovoljen, a trenutačni korak nije završen, tada se postavlja izlaz #SEQ.SU.sPrgRelease[7], koji omogućuje komunikaciju s dovodom smjese prema liniji. Zatim se provodi drugi dio provjere, gdje se također kontrolira uvjet "StepEnd". Ako korak nije završen, postavlja se izlaz #SEQ.SU.sPrgRelease[15], koji aktivira komunikaciju povrat tekućine iz linije natrag u tank ili u odvod.

### **3.2.5. Korak 5: Predrad linije (Preun line)**

Nakon završene faze odabira linije, sustav prelazi u fazu predrada line, koja služi za pripremu linije prije glavnog procesa čišćenja. Cijevi su već napunjene vodom iz prethodnog izvršavanja, a sada je potrebno zamijeniti tu vodu smjesom iz jednog od tankova s kemikalijama. Cilj ove faze je istisnuti preostalu vodu iz cijevi i pripremiti liniju kemijskom smjesom za daljnji proces. U ovoj fazi otvara se ventil na jednom od kemijskih tankova (kiselina ili lužina), dok pumpa potiskuje smjesu kroz cijevi prema odvodu. Smjesa istiskuje preostalu vodu, a senzori kontinuirano prate dva ključna uvjeta: provodljivost (conductivity) i količinu (amount) istisnute tekućine.

Faza istiskivanja završava kada sustav zabilježi da je provodljivost dosegla zadanu vrijednost, što potvrđuje prisutnost kemijske smjese u liniji i kada je istisnuta dovoljna količina tekućine. U ovoj fazi objekt (tank) šalje liniji zahtjev za aktivaciju pumpe i pritom određuje način rada pumpe. Način rada pumpe može biti fiksni način rada (Fix) ili kontrolirani način rada (Control), ovisno o konfiguraciji procesa.

Za ovaj korak je odabran fiksni način rada s postavljenim kapacitetom pumpe na 50%, što omogućuje stabilan i predvidljiv protok tijekom ispiranja. Odabir načina rada omogućuje se pomoću enumeracije definirane u Configuration Manageru. Operator ima mogućnost odabira između Fix i Control načina rada, a nakon odabira, kod automatski osigurava ispravnu primjenu odabrane postavke u procesu. Kod kontroliranog načina rada, pumpa bi bila povezana s PID regulatorom, koji bi prilagođavao protok prema zadanim parametrima, dok se kod fiksnog načina rada održava stalna postavljena brzina. Nakon aktivacije pumpe i otvaranja ventila, tekućina počinje protjecati kroz liniju. Sustav kontinuirano nadzire protok pomoću senzora protoka, dok povratni senzor provjerava provodljivost. Ako bilo koji od ovih parametara odstupa od zadanih vrijednosti, sustav



Number	Name	Designation	Replacement mode	Setpoint	Unit	Input mode
<b>Functions</b>						
1	Time	Time	Local	00:00:02	[d] h:m:s	Standard
2	Agitator on time	Agitator on time	Local	00:01:00	[d] h:m:s	Standard
3	Agitator off time	Agitator off time	Local	00:10:00	[d] h:m:s	Standard
4	Line amount	Line amount	Local	2.0	hl	Standard
5	Line conductivity	Line conductivity	Local	10	mS	Standard
6	Line temperature	Line temperature	Local	0.0	°C	Invisible
<b>General</b>						
1	RunTime	Runtime	Local	00:00:00	sec	Standard
1	MonTime	Monitoring time	Local	00:02:00	sec	Standard
<b>Setpoints</b>						
1	Pump control	Pump control	Local	Fix		Standard
2	Flow	Flow	Local	0.000		Invisible
3	Pump speed	Pump speed	Local	50.000		Standard
4	Heating control	Heating control	Local	Off		Standard
5	Temperature	Temperature	Local	0.0	°C	Standard
6	Heating valve position	Heating valve position	Local	0	%	Standard
7	Pushout supply	Pushout supply	Local	Fresh water		Standard
8	Pushout condition	Pushout condition	Local	Amount,Conductivity		Standard

**Slika 3.6.** Prikaz parametara iz procedure

automatski prekida fazu preruna i šalje signal za intervenciju.

Komunikacija između tanka i linije i dalje je omogućena putem klase SeqCom (klasa 169), koja omogućuje objektu da liniji šalje informacije o tome koje operacije treba izvršiti. FIFO sustav osigurava redosljednu obradu svih zahtjeva, sprječavajući sukobe i istodobne naredbe, čime se održava stabilnost i sinkronizacija rada.

Input language: English [English]	
General	
Value	Designation
0	Off
1	Fix
2	Control

**Slika 3.7.** Prikaz enumeracije iz Configuration Managera

```

// Kod za odabir načina rada pumpe
L    "DB_LiT_C160_SEQ_IF".extValue[1]
L    L#0
==D
=    #PumpOff
L    "DB_LiT_C160_SEQ_IF".extValue[1]
L    L#1
==D
=    #PumpFix
L    "DB_LiT_C160_SEQ_IF".extValue[1]
L    L#2
==D
=    #PumpCtrl

```

Ovaj dio koda definira način rada pumpe u fazi zagrijavanja (ali može biti primjenjiv i na druge faze). Logika se temelji na vrijednosti parametra koji je povezan na enumeraciju te se koristi za određivanje načina rada pumpe. Parametar može imati tri moguće vrijednosti: 0, 1 ili 2, pri čemu svaka vrijednost određuje specifičan način rada vidljivog sa slike 3.7.

### 3.2.6. Korak 6: Zagrijavanje (Heating)

Nakon što je faza predrada linije završena, sustav prelazi u fazu zagrijavanja (engl. heating). Cilj ove faze je podizanje temperature smjese na optimalnu razinu kako bi se omogućilo učinkovito čišćenje i dezinfekcija. Smjesa, koja je sada prisutna u liniji, počinje cirkulirati kroz sustav. Pumpa potiskuje smjesu prema grijaču, gdje se vrši kontrolirano zagrijavanje. Sustav otvara ventil za dovod pare, a temperatura se precizno regulira pomoću PID regulatora. Regulator nadzire dovod pare i prilagođava ga kako bi temperatura smjese bila u skladu sa zadanim vrijednostima.

Senzori temperature kontinuirano prate stanje tekućine i signaliziraju sustavu ako dođe do odstupanja. Ako temperatura padne ispod zadane vrijednosti, PID regulator povećava dotok pare, a u slučaju pregrijavanja, dotok se smanjuje. Upravljanje je omogućeno kao i za pumpu, preko enumeracije i programskog koda za odabir načina rada

koji je kao i kod pumpe fiksni ili kontrolirani. Smjesa tijekom cijelog procesa zagrijavanja cirkulira kroz liniju i vraća se u tank, čime se osigurava ravnomjerna distribucija topline i održavanje homogenosti smjese. Sustav također prati uvjete za završetak faze, uključujući postizanje zadane temperature i njezino održavanje u zadanom vremenskom okviru. Kada se postigne stabilna zadana temperatura, faza zagrijavanja završava, sustav tada prelazi u sljedeći korak sekvence.

### **3.2.7. Korak 7: Poslijeradnja linije (Postrun line)**

Nakon faze zagrijavanja, sustav prelazi u fazu poslijeradnja linija (engl. postrun line). Ova faza služi za ispiranje i čišćenje linije nakon što je smjesa već prošla kroz proces čišćenja objekta. Cilj "postrun" faze je ukloniti preostale kemikalije iz linije i pripremiti sustav za završno ispiranje vodom.

U ovom koraku linija se ispira vodom. Sustav otvara odgovarajući ventil, dok pumpa nastavlja potiskivati vodu kroz liniju prema tanku (ako je tank pun, prema odvodu). Senzori za protok i provodljivost nadziru cijeli proces kako bi se osiguralo da je smjesa učinkovito isprana. Ako provodljivost smjese padne ispod zadane granice, sustav potvrđuje da je čišćenje linije završeno.

Također, i u ovoj fazi je aktivna komunikacija između tanka i linije putem SeqCom klase, kojom objekt definira radni način pumpe, ventila i ostalih komponenata. Ovisno o konfiguraciji, "pushout" uvjet u ovoj fazi je provodljivost. Kada je taj uvjet zadovoljen, "postrun line" faza završava. Linija je tada spremna za prijelaz u završno ispiranje vodom, čime se dovršava ciklus CIP procesa i osigurava da je linija potpuno čista i bez zaostalih kemikalija.

### **3.2.8. Korak 8: Ispiranje linije (Flush line)**

Nakon završene faze postruna, sustav prelazi u fazu flush line, što se prevodi kao ispiranje linije. Cilj ove faze je potpuno uklanjanje preostale smjese i kemikalija iz cjevovoda pomoću čiste vode. Time se osigurava da su linije i ostale komponente sustava spremne za sljedeću uporabu ili novi ciklus rada.

U ovoj fazi sustav otvara ventil za dovod čiste vode, a pumpa potiskuje vodu kroz

liniju. Za razliku od prethodnih faza, gdje se pratio parametar provodljivosti, ovdje se praćenje ispiranja temelji isključivo na parametru količine tekućine. Sustav mjeri količinu vode izraženu u hektolitrima, a očitavanje se temelji na protoku kroz liniju, koji prati senzor protoka. Pumpa i ventili rade dok sustav ne zabilježi da je prošla zadana količina vode. Kada je ovaj uvjet ispunjen, faza ispiranja završava.

### **3.2.9. Korak 9: Oslobađanje linije (Release line)**

Nakon završetka faze ispiranja, sustav prelazi u fazu oslobađanja linije. Cilj ove faze je dovršiti CIP ciklus tako što se linija vraća u početno stanje mirovanja, spremna za sljedeće korištenje ili idući proces.

### **3.2.10. Korak 10: Servis (Service)**

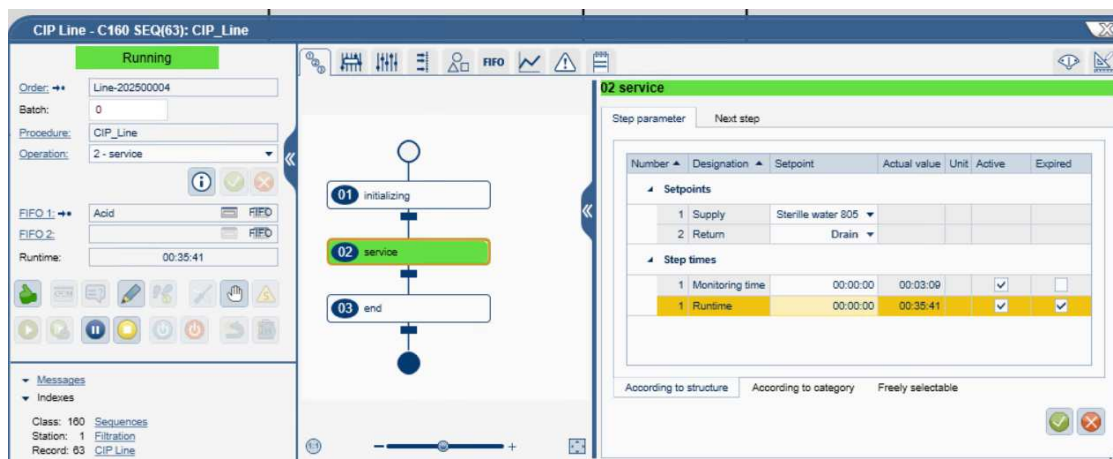
Cilj ove faze je osigurati da je sustav spreman za daljnju upotrebu i da je tank u optimalnom stanju za posluživanje smjese prema objektu kojem je potrebno pranje. U koraku servis, kod tankova s kemikalijama obavlja se provjera spremnosti i održavanje sustava, dok je kod tanka s vodom ovaj korak jednostavniji jer se u njemu odvija samo punjenje. Budući da ovaj tank nije kompleksan kao kemijski tankovi – jer ne sudjeluje u fazama miješanja ili zagrijavanja – on prolazi samo kroz tri ključna koraka: inicijalizacija, servis i kraj. Kada se tank isprazni tijekom rada, u koraku servis puni se čistom vodom, pripremajući se za daljnje korištenje u fazama ispiranja i opskrbe linija. U ovoj fazi, tankovi s kemikalijama također su napunjeni i spremni s odgovarajućom smjesom, bilo da je riječ o kiselinu ili lužini, ovisno o potrebi objekta koji zahtijeva pranje.

### **3.2.11. Korak 11: Kraj (End)**

Nakon završetka faze service, sustav prelazi u posljednju fazu sekvence – end ili kraj. Ova faza označava završetak cijelog CIP ciklusa i vraćanje sustava u potpuno pasivno stanje mirovanja (Idle). U ovom koraku svi ventili se zatvaraju, pumpe se isključuju, a parametri sekvence poput brojača, statusnih zastavica i aktivnih signala resetiraju se na početne vrijednosti. Senzori i kontrolni sustav više ne prate parametre jer proces više nije aktivan.

### 3.3. Sekvenca linije

Sekvenca linije u CIP procesu sastoji se od tri ključna koraka: inicijalizacija, servis i kraj.



Slika 3.8. Puni prikaz sekvence linije

#### 3.3.1. Korak 1: Inicijalizacija (Initialising)

U ovom koraku isto kao i kod tankova provodi se provjera stanja svih elemenata linije. Svi ventili moraju biti zatvoreni, pumpa isključena, a senzori provjereni kako bi se utvrdilo da nema zaostalog tlaka ili tekućine u cijevima. Također se resetiraju svi brojači i parametri vezani za sekvencu. Cilj ovog koraka je osigurati da linija započne proces iz stabilnog i sigurnog početnog stanja.

#### 3.3.2. Korak 2: Servis (Service)

U fazi servisa linija ostaje u stanju pripravnosti, spremna za primanje smjese iz pripremljenog tanka. U ovoj fazi sustav prati i održava parametre kao što su tlak i spremnost ventila i pumpe. Ako objekt zatraži pranje, linija aktivira odgovarajuće komponente kako bi omogućila ispravan protok smjese prema objektu. Ova faza traje sve dok linija više nije potrebna za rad, osiguravajući da sustav ostane u optimalnom stanju.

#### 3.3.3. Korak 3: Kraj (End)

Faza kraja označava završetak ciklusa rada linije. U ovom koraku svi ventili se zatvaraju, pumpa se isključuje, a sustav provodi završne provjere kako bi osigurao da u liniji nema

zaostale tekućine ili tlaka. Parametri sekvence se resetiraju na početne vrijednosti, a linija prelazi u stanje mirovanja (Idle), spremna za sljedeći ciklus ili ponovno aktiviranje.

### **3.4. Uloga kiseline i lužine u pranju DAW linije i filtera**

U pivarskoj industriji proces čišćenja i dezinfekcije opreme od ključne je važnosti za očuvanje kvalitete proizvoda i osiguranje higijenskih standarda. U tu svrhu koristi se CIP (Clean-In-Place) sustav, koji uključuje tankove za različite medije, tank za kiselinu i tank za lužinu. Svaka od tih kemikalija ima specifičnu funkciju u uklanjanju različitih vrsta nečistoća koje se nakupljaju tijekom proizvodnog ciklusa. Konkretno ova stanica se koristi za pranje DAW linija i filtera.

#### **3.4.1. DAW linija**

DAW (De-aerated Water) linija transportira dekarboniziranu i deoksigeniranu vodu koja se koristi u proizvodnji piva. Ova voda igra ključnu ulogu u procesima razrjeđivanja piva, ispiranja opreme te miješanja sirovina. Tijekom rada u DAW linijama mogu se nakupljati organski ostaci i mineralne naslage, što zahtijeva redovito i detaljno čišćenje.

#### **3.4.2. Filteri**

Filteri se koriste za uklanjanje čestica i taloga iz piva, osiguravajući njegovu bistrinu i stabilnost. Oni uklanjaju preostali kvasac, proteine i ostale nečistoće nakon fermentacije. Zbog konstantnog protoka tekućine kroz filtere, na njihovim se površinama često nakupljaju organski i anorganski talozi, koji se moraju redovito uklanjati kako bi filteri ostali funkcionalni.

#### **3.4.3. Uloga lužine u pranju**

Lužina, najčešće natrijev hidroksid (NaOH), koristi se za uklanjanje organskih nečistoća. Tijekom pranja DAW linije i filtera, lužina razgrađuje i uklanja ostatke kvasca, proteina, šećera i masti koji se nakupljaju tijekom proizvodnog procesa. Ovaj korak osigurava da se svi organski ostaci uklone prije daljnjih faza čišćenja.

#### **3.4.4. Uloga kiseline u pranju**

Kiselina, poput fosforne ili dušične kiseline, koristi se za uklanjanje anorganskih naslaga, poput kamenca i mineralnih taloga koji se stvaraju zbog tvrde vode. Nakon što lužina ukloni organske ostatke, kiselina se koristi za otapanje i uklanjanje tih anorganskih naslaga, osobito u filterima gdje se često nakupljaju mineralni talozi. Pranje kiselinom osigurava dugotrajnu funkcionalnost opreme.

#### **3.4.5. Kada se koriste kiselina i lužina**

Upotreba kiseline i lužine ovisi o stanju opreme i vrsti nečistoća koje se trebaju ukloniti. Lužina se koristi u većini redovnih ciklusa čišćenja jer organski ostaci čine glavni uzrok kontaminacije u pivarskoj proizvodnji. S druge strane, kiselina se koristi periodično, najčešće nakon nekoliko ciklusa čišćenja lužinom ili kada se primijeti nakupljanje anorganskih naslaga.

## 4. Prikupljanje i obrada podataka

Prikupljanje i obrada podataka ključni su procesi u suvremenim industrijskim sustavima, gdje pouzdanost i učinkovitost igraju presudnu ulogu. Pravovremeno praćenje stanja komponenata, procesa i senzorskih mjerenja omogućuje optimizaciju rada, sprječavanje kvarova i povećanje produktivnosti. U nastavku je opisano kako se podaci prikupljaju, obrađuju i prenose unutar integriranih sustava.

Prilikom prikupljanja podataka koriste se sustavi za upravljanje bazama podataka kako bi se omogućila sigurna i strukturirana pohrana informacija. Time se osigurava učinkovito rukovanje velikim količinama podataka te njihova obrada putem upita i procedura. Podaci su organizirani u odgovarajuće kategorije prema svojoj namjeni, primjerice: status komponenti, operativne aktivnosti ili senzorska mjerenja ključna za praćenje uvjeta rada.

Za automatizaciju obrade i pohrane podataka koriste se pohranjene procedure, koje omogućuju brzu izvedbu operacija poput unosa, ažuriranja i dohvaćanja informacija iz baze podataka. Ove procedure često koriste ulazne parametre, poput identifikacijskih brojeva, vremena trajanja operacija ili vrijednosti specifičnih mjerenja. Definirani uvjeti unutar procedura određuju hoće li se postojeći zapisi ažurirati ili će se kreirati novi, čime se osigurava dosljednost i točnost podataka – ključnih za donošenje odluka u stvarnom vremenu.

Komunikacija između kontrolnih uređaja (npr. PLC-a) i baze podataka ostvaruje se putem konfiguracijskih alata. Ključni parametri komunikacije uključuju pointere na podatke, memorijske adrese i broj bajtova koji se prenose. Na ovaj način omogućena je dvosmjerna razmjena podataka između baze i sustava za upravljanje procesima: sustav upravljanja može dohvatiti podatke iz baze, dok se istovremeno prikupljeni podaci s



terena pohranjuju u bazu radi analize.

Unutar PLC programa razvijaju se posebne funkcije za upravljanje slanjem podataka prema bazi. Ove funkcije se obično aktiviraju događajima u sustavu, primjerice na početku ili završetku operacije. U tom trenutku podaci se dohvaćaju iz memorije uređaja, obrađuju i prosljeđuju odgovarajućim procedurama u bazi podataka. Time se osigurava da su svi relevantni podaci dostupni u stvarnom vremenu za analizu i praćenje performansi.

Ovakav integrirani sustav omogućuje stabilnu i pouzdanu razmjenu podataka unutar industrijskih procesa. Prikupljeni podaci koriste se za nadzor rada, planiranje preventivnog održavanja i optimizaciju proizvodnih kapaciteta, što povećava učinkovitost i sigurnost cijelog sustava. Ovakav pristup omogućuje bolju kontrolu procesa te smanjuje rizik od neplaniranih zastoja i kvarova.

## **4.1. Dijagnostika na temelju prikupljenih podataka**

Kako bi se omogućila učinkovita dijagnostika sustava, razvijene su pohranjene procedure (engl. stored procedures) koje omogućuju kontinuirano praćenje rada i stanja komponenata na temelju prikupljenih podataka. Dijagnostički proces fokusiran je na analizu ključnih operativnih parametara radi pravovremenog otkrivanja nepravilnosti ili problema u radu sustava.

Prva pohranjena procedura prati stanje svake komponente pojedinačno, bilježeći operativne sate (engl. operating hours) i broj ciklusa prebacivanja (engl. switching cycles). Ova analiza omogućuje detaljan uvid u radni vijek komponenata, učestalost njihovog korištenja i opterećenje kojem su izložene. Na temelju ovih podataka moguće je predvidjeti potrebu za preventivnim održavanjem ili zamjenom određene komponente. Ako procedura detektira značajno povećanje broja ciklusa ili skraćene vremenske intervale između operacija, može se identificirati potencijalno oštećenje ili ubrzano trošenje komponente. Time se omogućuje pravovremena reakcija, čime se smanjuje rizik od neplaniranih kvarova i optimizira raspored održavanja.

Druga pohranjena procedura fokusira se na praćenje svih događaja otvaranja i zatvaranja ventila. Bez obzira na konkretni ventil, ova procedura bilježi vrijeme trajanja svake

operacije i omogućuje analizu dinamike rada sustava. Ako se utvrdi da određeni ventil zahtijeva dulje vrijeme za otvaranje ili zatvaranje nego što je uobičajeno, sustav može generirati upozorenje o mogućem kvaru ili potrebi za održavanjem. Kontinuirano praćenje ovih podataka omogućuje optimizaciju rada sustava i sprječavanje potencijalnih zastoja.

Kombinacijom ovih dviju procedura dobiva se cjelovit sustav dijagnostike. Procedura za praćenje komponenti omogućuje detaljan uvid u performanse pojedinih elemenata, dok procedura za praćenje ventila pruža globalni pregled aktivnosti sustava. Ovakav integrirani pristup omogućuje ne samo pravovremenu reakciju na probleme već i dugoročnu optimizaciju i poboljšanje rada cijelog sustava. Za potrebe dijagnostike analizirani su podaci prikupljeni tijekom jednog mjeseca, čime se dobio detaljan uvid u stanje komponenata i njihovu učinkovitost. Fokus je bio na ključnim operativnim parametrima, poput radnih sati i broja ciklusa prebacivanja, kako bi se identificirali obrasci rada i potencijalni rizici od kvarova. Ovaj sustav dijagnostike osigurava veću pouzdanost i trajnost industrijskih sustava, smanjuje rizike od neočekivanih kvarova te doprinosi povećanju produktivnosti i optimizaciji resursa.

#### **4.1.1. Status komponente**

Razvijena je pohranjena procedura pod nazivom FER\_sp\_ComponentStatus 4.9. Ova procedura omogućuje ažuriranje i unos podataka o radu komponenata na temelju informacija prikupljenih iz kontrolnih sustava, poput PLC-a. Kroz ovu proceduru prati se status svake komponente, omogućavajući analizu performansi i pravovremeno održavanje.

Procedura prima nekoliko važnih ulaznih parametara, uključujući identifikacijske brojeve klase, PLC stanice i zapisa. Osim toga, bilježe se ključni operativni parametri poput broja ciklusa prebacivanja i ukupnih radnih sati. Ovi podaci omogućuju precizno praćenje koliko je puta određena komponenta, primjerice ventil, otvorena i zatvorena te koliko dugo je bila u funkciji. Na temelju tih informacija moguće je planirati održavanje i zamjenu dijelova prije nego što dođe do neplaniranih zastoja ili kvarova.

Nakon što su podaci poslani proceduri, prvo se automatski generira vremenski zapis trenutka obrade, a zatim se dohvaćaju dodatni podaci o nazivu i opisu komponente iz povezanih tablica baze podataka. Ovaj korak osigurava da su podaci o svakoj komponenti

```

USE [dbIdc]
GO

/***** Object: StoredProcedure [dbo].[FER_sp_ComponentStatus]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[FER_sp_ComponentStatus]
    @ClassNo          INT = -1,
    @PLCNo            INT = -1,
    @RecNo            INT = -1,
    @SwitchingCycle   INT = -1,
    @OperatingHours    INT = -1
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @TimeStamp AS DATETIME = GETUTCDATE();
    DECLARE @Name AS NVARCHAR(32) = N'';
    DECLARE @Designation AS NVARCHAR(255) = N'';

    SELECT TOP 1
        @Name = X.[szName]
        ,@Designation = L.[szText]
    FROM [dbIdc].[dbo].[tblItpObjectRef]
    AS R
    LEFT JOIN [dbIdc].[dbo].[tblICPDataX]
    AS X ON X.[nKey] = R.[nDataXLink]
    LEFT JOIN [dbIdc].[dbo].[tblIdcLanguageText]
    AS L ON L.[nKey] = X.[nLanguageTextLink]
    AND L.[nLanguageId] = 9

    WHERE
        R.[nClassNo] = @ClassNo
        AND R.[nStationNo] = @PLCNo
        AND R.[nRecordNo] = @RecNo;

MERGE [dbIdc].[dbo].[FER_tbl_ComponentStatus] AS Target
USING (
    SELECT
        @TimeStamp AS LastUpdate
        ,@ClassNo AS ClassNo
        ,@PLCNo AS PLCNo
        ,@RecNo AS RecNo
        ,@SwitchingCycle AS SwitchingCycle
        ,@OperatingHours AS OperatingHours
    ) AS Source
ON
    Target.[ClassNo] = Source.ClassNo
AND Target.[PLCNo] = Source.PLCNo
AND Target.[RecNo] = Source.RecNo

WHEN MATCHED THEN
    UPDATE SET
        [Name] = @Name
        ,[Designation] = @Designation
        ,[SwitchingCycle] = @SwitchingCycle
        ,[OperatingHours] = @OperatingHours
        ,[LastUpdate] = @TimeStamp

WHEN NOT MATCHED BY TARGET THEN
    INSERT (
        [ClassNo]
        ,[PLCNo]
        ,[RecNo]
        ,[Name]
        ,[Designation]
        ,[SwitchingCycle]
        ,[OperatingHours]
        ,[LastUpdate]
    ) VALUES (
        @ClassNo
        ,@PLCNo
        ,@RecNo
        ,@Name
        ,@Designation
        ,@SwitchingCycle
        ,@OperatingHours
        ,@TimeStamp
    );
END;
GO

```

(a)

(b)

**Slika 4.1.** Procedura za status komponenti

upotpunjeni i točno definirani. Dohvaćeni podaci uključuju naziv komponente koji se koristi za identifikaciju, kao i opis koji daje dodatne tehničke informacije.

Za pohranu svih ključnih informacija o radu pojedinih komponenata kreirana je tablica FER\_tbl\_ComponentStatus 4.2. Ova tablica predstavlja temelj za praćenje performansi sustava i dijagnostičke analize.

Na slici 4.3. nalazi se pregled konfiguracijskih parametara vezanih uz komunikaciju između PLC-a i baze podataka. Ovi parametri definiraju način na koji se podaci razmjenjuju, uključujući pointer na naredbe, status i greške. Ključna polja uključuju "Pointer to order trigger", "Pointer to status" i "Pointer to error", koji pokazuju na određene adrese u memoriji. Ovdje se, primjerice, koristi adresa DB5001.DBX6.0 za pokretanje naredbi, dok su za status i greške dodijeljene adrese DB5001.DBB7 i DB5001.DBB8. Definiran je i broj stanice servera (Server station number), koji ima vrijednost 101, kao i tip telegrama (Telegram type), postavljen na vrijednost 128. Dodatno su specificirani brojevi baza podataka i broj bajtova za izvorne i ciljane podatke, gdje izvorna baza ima broj 5001, a

```

USE [dbIdc]
GO

/***** Object: Table [dbo].[FER_tbl_ComponentStatus]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[FER_tbl_ComponentStatus](
    [ClassNo] [int] NOT NULL,
    [PLCNo] [int] NOT NULL,
    [RecNo] [int] NOT NULL,
    [Name] [nvarchar](32) NULL,
    [Designation] [nvarchar](255) NULL,
    [SwitchingCycle] [int] NULL,
    [OperatingHours] [int] NULL,
    [LastUpdate] [datetime] NULL,
    CONSTRAINT [PK_FER_tbl_ComponentStatus] PRIMARY KEY CLUSTERED
    (
        [ClassNo] ASC,
        [PLCNo] ASC,
        [RecNo] ASC
    )
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

```

**Slika 4.2.** Tablica za status komponenti

prijenos obuhvaća 18 bajtova podataka. Vrijeme odziva ili timeout definirano je na 60 sekundi, čime se osigurava stabilnost komunikacije i pravovremeno izvršenje naredbi. Ovi parametri omogućuju usklađenu razmjenu informacija između baze podataka i PLC-a, što je ključno za praćenje i kontrolu industrijskih procesa.

Property	Value (DB)	Value (PLC)	
Object record valid		Yes	
Object locked	No	No	
Allocation of the status	released	released	
Allocation of the error	released	released	
Mode	No response expected	No response expected	
Pointer to order trigger	DB5001.DBX6.0	DB5001.DBX6.0	
Pointer to status	DB5001.DBB7	DB5001.DBB7	
Pointer to error	DB5001.DBW8	DB5001.DBW8	
Server station number	101	101	
Service	128	128	
Telegram type	65	65	
Source: DB number	5001	5001	
Source: As of byte	10	10	
Source: Number of bytes	18	18	
Target: DB number	0	0	
Target: As of byte	0	0	
Target: Number of bytes	0	0	
Timeout	60	60	sec

**Slika 4.3.** Konfiguracija parametara za komunikaciju baze i PLC-a

Funkcija FC5001 koristi se za upravljanje cikličkim slanjem podataka iz PLC-a prema bazi podataka. Uloga varijable UserTimer, koja se povećava svakih 1000 milisekundi, ključna je za regulaciju učestalosti prijensa. Funkcija provjerava stanje timera svakih 5 sekundi, koristeći tu vrijednost kao prag za slanje. Kada je uvjet zadovoljen, funkcija prenosi podatke poput identifikacijskih brojeva, radnih sati i broja ciklusa prebacivanja u

komunikacijski objekt DBCom\_ComponentStatus. Nakon prijena podataka, upravlja se statusnim signalima kako bi se potvrdila uspješnost ili greška u komunikaciji.

\\FC5001

```
L      #Queue
+      -1
L      5
*I
L      "DBCom_ComponentStatus".UserTimer
==I
A(
L      #Queue
L      0
>I
)
=      #Send

A      #Send
JCN   snd
L      #ClassNo
T      "DBCom_ComponentStatus".Send.ClassNo
L      "DB_System".ThisStation
T      "DBCom_ComponentStatus".Send.PLCNo
L      #RecNo
T      "DBCom_ComponentStatus".Send.RecNo
L      #OperatingHours
T      "DBCom_ComponentStatus".Send.OperatingHours
L      #SwitchingCycles
T      "DBCom_ComponentStatus".Send.SwitchingCycles
snd:  NOP  0

AN    "DBCom_ComponentStatus".Comm.OrderInitiation
A      #Send
```

```

R      "DBCom_ComponentStatus".Comm.sReadyOK
R      "DBCom_ComponentStatus".Comm.sReadyErr
S      "DBCom_ComponentStatus".Comm.OrderInitiation

A      "DBCom_ComponentStatus".Comm.sReadyOK
O      "DBCom_ComponentStatus".Comm.sReadyErr
AN     "DBCom_ComponentStatus".Comm.sRunning
AN     #Send
R      "DBCom_ComponentStatus".Comm.OrderInitiation

```

Ovaj dio koda ispod koristi 1SecPuls za povećavanje UserTimer svakih 1000 milisekundi. Kada UserTimer dostigne ili premaši vrijednost definiranu u UserCycle, timer se resetira. Nakon što se ispune uvjeti za cikličko slanje, poziva se funkcija Send\_ComponentStatus. Ovaj mehanizam osigurava pravovremeno i sinkronizirano slanje podataka prema bazi. Konkretno je ovdje napravljeno za ventil iz klase 25, rednog broja 145, a analogno vrijedi za sve ostale.

\\FC889 Zyklus ende

```

A      "1SecPuls"
JCN   ncnt
L      "DBCom_ComponentStatus".UserTimer
+      1
L      "DBCom_ComponentStatus".UserCycle
>=I
TAK
JCN   n1
L      0
n1:   T      "DBCom_ComponentStatus".UserTimer
ncnt: NOP  0

CALL  "Send_ComponentStatus"
Queue      :=1
ClassNo    :=25

```

```
RecNo           :=145
SwitchingCycles:="DB_DiT_C25_S SZ".DCM[145].sActOperatingCycle
OperatingHours  :="DB_DiT_C25_S SZ".DCM[145].sActOperatingHours
```

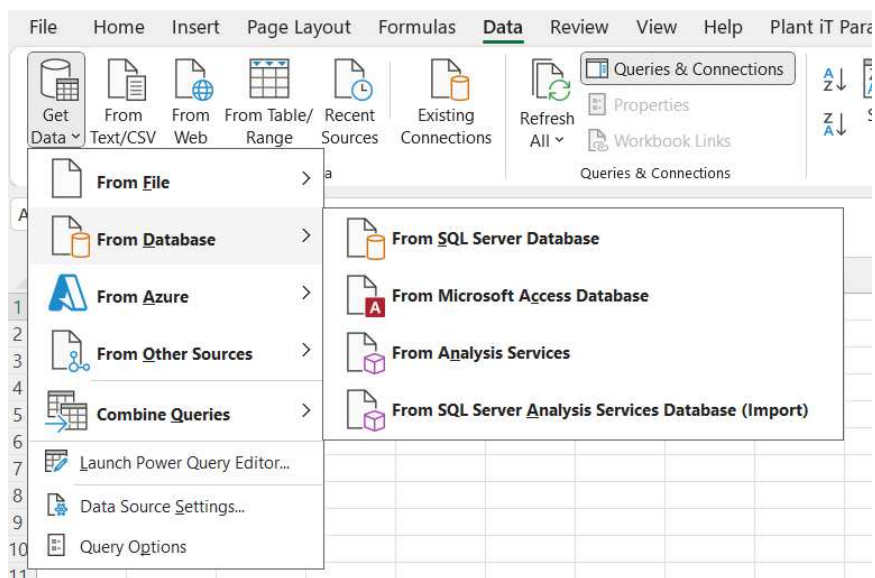
Na slici 4.4. je prikazan dio korisničkog sučelja u programu Microsoft Excel, točnije kartica Data s opcijama za povezivanje i uvoz podataka. Ova funkcionalnost omogućuje korisnicima dohvaćanje podataka iz različitih izvora, uključujući tekstualne datoteke, baze podataka i vanjske sustave. Pod opcijom Get Data nalaze se različiti izvori iz kojih se podaci mogu uvesti.

Korištena je opcija From SQL Server Database. Opcija From SQL Server Database u programu Excel omogućuje povezivanje s bazom podataka koja se nalazi na Microsoft SQL Serveru, što je posebno korisno za dohvaćanje i analizu velikih količina podataka. Ova funkcionalnost namijenjena je korisnicima koji rade s centraliziranim bazama podataka u poslovnom okruženju, gdje je potrebno brzo i jednostavno preuzeti podatke za daljnju obradu. Nakon odabira opcije, korisniku se otvara čarobnjak koji ga vodi kroz proces povezivanja s bazom. Prvi korak uključuje unos osnovnih podataka o serveru, poput naziva ili IP adrese SQL Servera te naziva baze iz koje se podaci trebaju dohvatiti. Kako bi se osigurala sigurnost pristupa, korisnik mora odabrati vrstu autentifikacije. Dostupne su dvije mogućnosti: Windows Authentication, koja koristi postojeći korisnički račun unutar Windows sustava, ili SQL Server Authentication, gdje je potrebno ručno unijeti korisničko ime i lozinku. Nakon uspješnog povezivanja, korisnik može pregledati strukturu baze podataka, uključujući tablice i view-ove. Iz ponuđenih elemenata moguće je odabrati podatke koji će se uvesti u Excel.

Pritom Excel nudi opcije za prilagodbu i filtriranje podataka prije uvoza, koristeći alat Power Query Editor. Ovim alatom moguće je transformirati podatke, primijeniti različite filtere, grupirati informacije i prilagoditi ih potrebama analize. Nadalje, Excel pruža mogućnost kombiniranja različitih izvora podataka putem opcije Combine Queries, što je korisno za integraciju i analizu velikih setova podataka iz različitih izvora.

Na temelju prikazanih podataka, posebnu pažnju privlače tri ventila zbog specifičnih karakteristika koje se odnose na broj ciklusa prebacivanja i operativne sate.

Prvi ventil, Drain valve (V\_CL\_D), ima visok broj switching cycle vrijednosti, koja



Slika 4.4. Dohvat podataka iz SQL tablice

	A	B	C	D	E	F	G	H
1	ClassNo	PLCNo	RecNo	Name	Designation	SwitchingCycle	OperatingHours	LastUpdate
2	25	1	160	V_CL_D	Drain valve	960	58	1/28/2025 23:59
3	25	1	287	V_CTW_W_S	Fresh water supply	960	45	1/28/2025 23:59
4	25	1	153	V_CL_CTA_S	Acid supply valve	128	8	1/28/2025 23:59
5	25	1	154	V_CL_CTA_R	Acid return valve	128	9	1/28/2025 23:59
6	25	1	155	V_CTA_W_S	Fresh water supply valve	192	7	1/28/2025 23:59
7	25	1	145	V_CL_B	Bypass valve	320	53	1/28/2025 23:59
8	25	1	820	V_CL_S_S	Steam supply valve	320	13	1/28/2025 23:59
9	25	1	901	V_CL_S_CVR	Steam supply control valve release	320	13	1/28/2025 23:59
10	25	1	152	V_CL_CTW_S	Fresh water supply valve	320	27	1/28/2025 23:58
11	26	1	1	V_CL_C	Circulation valve	320	13	1/28/2025 23:59
12	25	1	156	V_CL_CTC_S	Caustic supply valve	512	32	1/28/2025 23:59
13	25	1	157	V_CL_CTC_R	Caustic return valve	512	36	1/28/2025 23:59
14	25	1	158	V_CTC_W_S	Fresh water supply valve	128	5	1/28/2025 23:59

Slika 4.5. Prikaz podataka iz Excel tablice

iznosi 960. Ova brojka ukazuje na intenzivnu učestalost otvaranja i zatvaranja, što može značiti da je ventil pod velikim operativnim opterećenjem. Sličnu karakteristiku ima i ventil Fresh water supply (V\_CTW\_W\_S), koji također bilježi 960 ciklusa, što ga čini važnim za praćenje.

S druge strane, ventil Bypass valve (V\_CL\_B) izdvaja se zbog visokog broja operating hours, koji iznosi 53 sata. Iako broj ciklusa ovog ventila iznosi 320, duže vrijeme rada može ukazivati na to da ventil ostaje otvoren ili zatvoren kroz duže periode, što može predstavljati specifičan način rada.

#### 4.1.2. Vrijeme odziva aktuatora

Razvijena je procedura pod nazivom FER\_sp\_ComponentTraveling 4.6. Ova procedura omogućuje praćenje i unos podataka o vremenima odziva aktuatora na temelju informa-



cija koje se prikupljaju iz kontrolnih sustava poput PLC-a. Praćenjem vremena i smjera kretanja svake komponente omogućava se pravodobna analiza rada i performansi te prevencija kvarova koji mogu nastati uslijed nepravilnosti u radu.

Procedura prima nekoliko važnih ulaznih parametara, uključujući identifikacijske brojeve klase, PLC stanice i zapisa. Osim toga, ključni parametri uključuju smjer kretanja i trajanje kretanja. Ovi podaci omogućuju praćenje koliko dugo i u kojem smjeru se komponenta, primjerice ventil, otvarala ili zatvarala. Na temelju tih informacija moguće je planirati preventivno održavanje te smanjiti rizik od zastoja ili oštećenja komponenata.

```

USE [dbIdc]
GO

/***** Object: StoredProcedure [dbo].[FER_sp_ComponentTraveling]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[FER_sp_ComponentTraveling]
    @ClassNo          INT = -1,
    @PLCNo            INT = -1,
    @RecNo            INT = -1,
    @TravelingDirection INT = -1,
    @TravelingTime     INT = -1
AS
BEGIN

    SET NOCOUNT ON;

    DECLARE @TimeStamp AS DATETIME = GETUTCDATE();

    DECLARE @Name AS NVARCHAR(32) = N'';
    DECLARE @Designation AS NVARCHAR(255) = N'';
    DECLARE @Direction AS NVARCHAR(32) = N'';

    SELECT TOP 1
        @Name = X.[szName]
        ,@Designation = L.[szText]
    FROM [dbIdc].[dbo].[tblItpObjectRef] AS R
    LEFT JOIN [dbIdc].[dbo].[tblICDataX] AS X ON X.[nkey] = R.[nDataLink]
    LEFT JOIN [dbIdc].[dbo].[tblIdcLanguageText] AS L ON L.[nkey] = X.[nLanguageTextLink]
    AND L.[nLanguageId] = 9

    WHERE
        R.[nClassNo] = @ClassNo
        AND R.[nStationNo] = @PLCNo
        AND R.[nRecordNo] = @RecNo;

    SET @Direction =
        CASE
            WHEN @TravelingDirection = 1 THEN 'Opening'
            WHEN @TravelingDirection = 2 THEN 'Closing'
            ELSE 'Undefined'
        END;

    INSERT INTO [dbIdc].[dbo].[FER_tbl_ComponentTraveling] ([ClassNo], [PLCNo], [RecNo],
        [Name], [Designation], [TravelingDirection], [TravelingTime], [Lastupdate])
    VALUES (@ClassNo, @PLCNo, @RecNo, @Name, @Designation, @Direction, @TravelingTime, @TimeStamp);

END
GO
    
```

**Slika 4.6.** Izrada procedure za vrijeme odziva aktuatora

Kada se procedura aktivira, automatski se generira vremenski zapis trenutka obrade, a potom se dohvaćaju dodatni podaci o nazivu i opisu komponente iz povezanih tablica baze podataka. Na temelju ulaznog parametra @TravelingDirection, smjer kretanja se definira kao "Opening" (otvaranje) ili "Closing" (zatvaranje). Zatim se svi prikupljeni podaci pohranjuju u tablicu FER\_tbl\_ComponentTraveling 4.7.

Tablica FER\_tbl\_ComponentTraveling kreirana je kako bi pohranila sve ključne informacije o vremenu odziva pojedinih komponenata, uključujući identifikacijske brojeve, naziv, opis, smjer kretanja, vrijeme odziva te vrijeme zadnjeg ažuriranja. Struktura tablice omogućuje detaljno praćenje rada komponenata te analizu vremenskih obrazaca njihovog otvaranja i zatvaranja.

Na slici 4.8. prikazana je konfiguracija parametara koji su vezani uz komunikaciju

```

USE [dbIdc]
GO

/***** Object: Table [dbo].[FER_tbl_ComponentTraveling]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FER_tbl_ComponentTraveling](
    [UID] [int] IDENTITY(1,1) NOT NULL,
    [ClassNo] [int] NOT NULL,
    [PLCNo] [int] NOT NULL,
    [RecNo] [int] NOT NULL,
    [Name] [nvarchar](32) NULL,
    [Designation] [nvarchar](255) NULL,
    [TravelingDirection] [nvarchar](32) NULL,
    [TravelingTime] [int] NULL,
    [LastUpdate] [datetime] NULL,
    CONSTRAINT [PK_FER_tbl_ComponentTraveling] PRIMARY KEY CLUSTERED
    (
        [UID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

```

**Slika 4.7.** Izrada tablice za vrijeme odziva aktuatora

između PLC-a i baze podataka za ovu funkcionalnost. Parametri uključuju pointere na naredbe, status i greške, koji definiraju način razmjene podataka između PLC-a i baze. Primjer adrese koja se koristi za iniciranje naredbi je DB5002.DBX0.0, dok su status i greške vezani uz adrese DB5002.DBB1 i DB5002.DBW2. Za ovu komunikaciju definirana je vrijednost broja servera kao 101, a tip telegrama je postavljen na 128. Prijenos podataka sada obuhvaća 14 bajtova s izvorne baze podataka broj 5002, a vrijeme odziva (timeout) iznosi 60 sekundi.

Property	Value (DB)	Value (PLC)	
Object record valid		Yes	
Object locked	No	No	
Allocation of the status	released	released	
Allocation of the error	released	released	
Mode	No response expected	No response expected	
Pointer to order trigger	DB5002.DBX0.0	DB5002.DBX0.0	
Pointer to status	DB5002.DBB1	DB5002.DBB1	
Pointer to error	DB5002.DBW2	DB5002.DBW2	
Server station number	101	101	
Service	128	128	
Telegram type	66	66	
Source: DB number	5002	5002	
Source: As of byte	4	4	
Source: Number of bytes	14	14	
Target: DB number	0	0	
Target: As of byte	0	0	
Target: Number of bytes	0	0	
Timeout	60	60	sec

**Slika 4.8.** Konfiguracija parametara za komunikaciju baze i PLC-a

Funkcija FC5002 služi za upravljanje cikličkim prikupljanjem i slanjem podataka o vremenima odziva aktuatora prema bazi podataka. Na početku funkcije definiraju se

osnovne informacije o komponenti, uključujući identifikacijski broj klase i redni broj zapisa. Vrijednosti početnih parametara za smjer odziva i trajanje kretanja postavljaju se na nulu. U ovom stanju varijabla #Send, koja označava spremnost za slanje podataka, inicijalno je resetirana. Nakon inicijalizacije provjerava se stanje memorijskih zastavica koje definiraju status odziva. Ako je aktivna zastavica koja označava da je komponenta započela otvaranje, varijabla #Send se postavlja na aktivno stanje. U tom trenutku, smjer kretanja definira se kao "Opening", a vrijeme trajanja otvaranja dohvaća se iz odgovarajuće memorijske lokacije i prenosi u varijablu #TravelingTime.

Ako je otvaranje završeno, funkcija provjerava zastavicu zatvaranja. Ako je aktivna zastavica koja označava početak zatvaranja, ponovno se aktivira varijabla #Send, smjer kretanja definira se kao "Closing", a vrijeme trajanja zatvaranja pohranjuje se u #TravelingTime. Ovim koracima funkcija osigurava da se pravilno bilježe oba moguća smjera kretanja komponente.

Nakon definiranja stanja i parametara vremena odziva, funkcija provjerava je li aktivan uvjet za slanje podataka. Ako uvjet nije zadovoljen, funkcija preskače sljedeće korake. Međutim, ako je #Send aktivan, podaci se pripremaju za slanje. Prvo se prenose identifikacijski brojevi komponente i PLC stanice u odgovarajuće polja objekta za komunikaciju DBCom\_ComponentTraveling. Zatim se prenose podaci o smjeru i trajanju vremena odziva.

U sljedećem koraku funkcija ažurira memorijske zastavice kako bi označila da su podaci uspješno proslijeđeni. Ako je smjer kretanja bio "Opening", zastavica OpenMem se resetira. Slično tome, ako je smjer bio "Closing", resetira se zastavica CloseMem.

Nakon što su podaci pripremljeni, funkcija inicira proces slanja. Prvo se brišu eventualne prethodne oznake o uspješnosti ili grešci u komunikaciji, a zatim se postavlja naredba za pokretanje slanja. Kada je slanje započeto, funkcija prati stanje signala koji potvrđuju uspješnost ili grešku u komunikaciji. Ako dođe do pogreške ili slanje nije moguće, naredba za pokretanje se resetira.

Konačno, funkcija ažurira vrijednosti vremena odziva komponente. Vrijeme otvaranja izračunava se kao razlika između trenutka završetka i trenutka početka otvaranja, nakon čega se vrijednost množi faktorom 100 i pohranjuje u komunikacijski objekt. Slično

se izračunava i vrijeme zatvaranja. Na temelju ovih izračuna funkcija ažurira memorijske zastavice koje označavaju trenutačna stanja otvaranja i zatvaranja komponente.

The image shows two SQL scripts side-by-side in a code editor. Script (a) is a T-SQL procedure named 'FER\_sp\_ComponentStatus' that sets ANSI\_NULLS and QUOTED\_IDENTIFIER, then declares variables for timestamp, name, and designation. It uses a SELECT TOP 1 query to fetch component details from a table linked to a language table. Script (b) is a MERGE statement that updates the 'FER\_tbl\_ComponentStatus' table with data from a source table, matching on ClassNo, PLCNo, and RecNo. It includes UPDATE and INSERT clauses for matched and non-matched rows respectively.

```

USE [dbIdc]
GO

/***** Object: StoredProcedure [dbo].[FER_sp_ComponentStatus]
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[FER_sp_ComponentStatus]
    @ClassNo INT = -1,
    @PLCNo INT = -1,
    @RecNo INT = -1,
    @SwitchingCycle INT = -1,
    @OperatingHours INT = -1
AS
BEGIN

    SET NOCOUNT ON;

    DECLARE @TimeStamp AS DATETIME = GETUTCDATE();
    DECLARE @Name AS NVARCHAR(32) = N'';
    DECLARE @Designation AS NVARCHAR(255) = N'';

    SELECT TOP 1
        @Name = X.[szName]
        ,@Designation = L.[szText]
    FROM [dbIdc].[dbo].[tblItpObjectRef]
    AS R
    LEFT JOIN [dbIdc].[dbo].[tblCPDataX]
    AS X ON X.[nKey] = R.[nDataXLink]
    LEFT JOIN [dbIdc].[dbo].[tblIdcLanguageText]
    AS L ON L.[nKey] = X.[nLanguageTextLink]
    AND L.[nLanguageId] = 9

    WHERE
        R.[nClassNo] = @ClassNo
        AND R.[nStationNo] = @PLCNo
        AND R.[nRecordNo] = @RecNo;

MERGE [dbIdc].[dbo].[FER_tbl_ComponentStatus] AS Target
USING (
    SELECT
        @TimeStamp AS LastUpdate
        ,@ClassNo AS ClassNo
        ,@PLCNo AS PLCNo
        ,@RecNo AS RecNo
        ,@SwitchingCycle AS SwitchingCycle
        ,@OperatingHours AS OperatingHours
    ) AS Source
ON
    Target.[ClassNo] = Source.ClassNo
AND Target.[PLCNo] = Source.PLCNo
AND Target.[RecNo] = Source.RecNo

WHEN MATCHED THEN
    UPDATE SET
        [Name] = @Name
        ,[Designation] = @Designation
        ,[SwitchingCycle] = @SwitchingCycle
        ,[OperatingHours] = @OperatingHours
        ,[LastUpdate] = @TimeStamp

WHEN NOT MATCHED BY TARGET THEN
    INSERT (
        [ClassNo]
        ,[PLCNo]
        ,[RecNo]
        ,[Name]
        ,[Designation]
        ,[SwitchingCycle]
        ,[OperatingHours]
        ,[LastUpdate]
    ) VALUES (
        @ClassNo
        ,@PLCNo
        ,@RecNo
        ,@Name
        ,@Designation
        ,@SwitchingCycle
        ,@OperatingHours
        ,@TimeStamp
    );

END;
GO
    
```

Slika 4.9. Procedura za status komponenti

```

\\FC5002

// Basic info

L      25

T      #ClassNo

L      145

T      #RecNo

L      0

T      #TravelingDirection

L      0

T      #TravelingTime

CLR

=      #Send
    
```

```

// Determine what to send
    A    "DBCom_ComponentTraveling".DBCom[2].Aux.OpenMem
    JCN  n02x
    =    #Send
    L    1
    T    #TravelingDirection
    L    "DBCom_ComponentTraveling".DBCom[2].Aux.OpeningTime
    T    #TravelingTime
n02x: NOP  0

    AN   "DBCom_ComponentTraveling".DBCom[2].Aux.OpenMem
    A    "DBCom_ComponentTraveling".DBCom[2].Aux.CloseMem
    JCN  n02y
    =    #Send
    L    2
    T    #TravelingDirection
    L    "DBCom_ComponentTraveling".DBCom[2].Aux.ClosingTime
    T    #TravelingTime
n02y: NOP  0

// Prepare data for sending
    A    #Send
    JCN  n02s
    L    #ClassNo
    T    "DBCom_ComponentTraveling".DBCom[2].Send.ClassNo
    L    "DB_System".ThisStation
    T    "DBCom_ComponentTraveling".DBCom[2].Send.PLCNo
    L    #RecNo
    T    "DBCom_ComponentTraveling".DBCom[2].Send.RecNo
    L    #TravelingDirection
    T    "DBCom_ComponentTraveling".DBCom[2].Send.TravelingDirection

```

```

L    #TravelingTime
T    "DBCom_ComponentTraveling".DBCom[2].Send.TravelingTime

L    #TravelingDirection
L    1
==I
R    "DBCom_ComponentTraveling".DBCom[2].Aux.OpenMem
TAK
L    2
==I
R    "DBCom_ComponentTraveling".DBCom[2].Aux.CloseMem
n02s: NOP    0

// Trigger sending

AN   "DBCom_ComponentTraveling".DBCom[2].Comm.OrderInitiation
A    #Send
R    "DBCom_ComponentTraveling".DBCom[2].Comm.sReadyOK
R    "DBCom_ComponentTraveling".DBCom[2].Comm.sReadyErr
S    "DBCom_ComponentTraveling".DBCom[2].Comm.OrderInitiation

A    "DBCom_ComponentTraveling".DBCom[2].Comm.sReadyOK
O    "DBCom_ComponentTraveling".DBCom[2].Comm.sReadyErr
AN   "DBCom_ComponentTraveling".DBCom[2].Comm.sRunning
AN   #Send
R    "DBCom_ComponentTraveling".DBCom[2].Comm.OrderInitiation

// Memorize state
L    "DB_DiT_C25_ExtRecord".DCM[145].sTravelTime_open_finish
L    "DB_DiT_C25_ExtRecord".DCM[145].sTravelTime_open_start
-I

```

```

L      100
*I
T      "DBCom_ComponentTraveling".DBCom[2].Aux.OpeningTime

L      "DB_DiT_C25_ExtRecord".DCM[145].sTravelTime_close_finish
L      "DB_DiT_C25_ExtRecord".DCM[145].sTravelTime_close_start
-I
L      100
*I
T      "DBCom_ComponentTraveling".DBCom[2].Aux.ClosingTime

A      "DB_DiT_C25_ExtRecord".DCM[145].Opening
FN     "DBCom_ComponentTraveling".DBCom[2].Aux.OpenAux
S      "DBCom_ComponentTraveling".DBCom[2].Aux.OpenMem

A      "DB_DiT_C25_ExtRecord".DCM[145].Closing
FN     "DBCom_ComponentTraveling".DBCom[2].Aux.CloseAux
S      "DBCom_ComponentTraveling".DBCom[2].Aux.CloseMem

```

Na isti način, za drugu tablicu FER\_tbl\_ComponentTraveling, koristi se opcija From SQL Server Database u programu Microsoft Excel. Nakon što korisnik odabere ovu opciju, čarobnjak ga vodi kroz proces povezivanja s bazom podataka. Unose se podaci o serveru, poput naziva ili IP adrese SQL Servera te ime baze iz koje se dohvaćaju podaci o vremenu odziva komponenti. Kao i kod prve tablice, korisnik odabire način autentifikacije, pri čemu može birati između Windows Authentication ili SQL Server Authentication. Nakon što je veza s bazom uspostavljena, korisnik može pregledati strukturu baze i odabrati tablicu FER\_tbl\_ComponentTraveling za dohvaćanje podataka.

U posljednjih mjesec dana provedena je detaljna analiza rada ključnih ventila u sustavu kako bi se dobio uvid u učestalost njihovih operacija, operativne sate te vrijeme potrebno za otvaranje i zatvaranje. Kao najvažniji ventili u sustavu istaknuli su se Drain valve, Fresh water supply valve i Bypass valve, koji su svojim brojem operacija i radnim satima pokazali ključnu ulogu u održavanju stabilnosti sustava. Drain valve zabilježio

UID	ClassNo	PLCNo	Rechno	Name	Designation	TravelingDirection	TravelingTime	LastUpdate
1	25	1	160	V_CL_D	Drain valve	Opening		2165 2024-12-28 04:19:35
2	25	1	287	V_CTW_W_S	Fresh water supply	Opening		2151 2024-12-28 04:19:35
3	25	1	287	V_CTW_W_S	Fresh water supply	Closing		2121 2024-12-28 04:26:05
4	25	1	153	V_CL_CTA_S	Acid supply valve	Opening		2111 2024-12-28 04:28:15
5	25	1	160	V_CL_D	Drain valve	Closing		2134 2024-12-28 04:29:15
6	25	1	154	V_CL_CTA_R	Acid return valve	Opening		2113 2024-12-28 04:29:16
7	25	1	153	V_CL_CTA_S	Acid supply valve	Closing		2087 2024-12-28 04:33:15
8	25	1	287	V_CTW_W_S	Fresh water supply	Opening		2151 2024-12-28 04:34:57
9	25	1	154	V_CL_CTA_R	Acid return valve	Closing		2089 2024-12-28 04:35:56
10	25	1	160	V_CL_D	Drain valve	Opening		2165 2024-12-28 04:35:57
11	25	1	287	V_CTW_W_S	Fresh water supply	Closing		2121 2024-12-28 04:40:57
12	25	1	160	V_CL_D	Drain valve	Closing		2134 2024-12-28 04:42:52
13	25	1	155	V_CTA_W_S	Fresh water supply valve	Opening		2032 2024-12-28 04:42:57
14	25	1	287	V_CTW_W_S	Fresh water supply	Opening		2151 2024-12-28 04:42:57
15	25	1	287	V_CTW_W_S	Fresh water supply	Closing		2121 2024-12-28 04:47:23
16	25	1	160	V_CL_D	Drain valve	Opening		2165 2024-12-28 04:47:28
17	25	1	145	V_CL_B	Bypass valve	Opening		2101 2024-12-28 04:47:28
18	25	1	153	V_CL_CTA_S	Acid supply valve	Opening		2111 2024-12-28 04:47:28
19	25	1	155	V_CTA_W_S	Fresh water supply valve	Closing		2007 2024-12-28 04:47:28
20	25	1	160	V_CL_D	Drain valve	Closing		2134 2024-12-28 04:52:28
21	25	1	154	V_CL_CTA_R	Acid return valve	Opening		2113 2024-12-28 04:52:28
22	25	1	820	V_CL_S_S	Steam supply valve	Opening		2089 2024-12-28 04:52:28
23	25	1	901	V_CL_S_CVR	Steam supply control valve release	Opening		2089 2024-12-28 04:52:28
24	25	1	820	V_CL_S_S	Steam supply valve	Closing		2064 2024-12-28 04:57:28
25	25	1	153	V_CL_CTA_S	Acid supply valve	Closing		2087 2024-12-28 04:57:28
26	25	1	152	V_CL_CTW_S	Fresh water supply valve	Opening		2093 2024-12-28 04:57:28
27	25	1	901	V_CL_S_CVR	Steam supply control valve release	Closing		2064 2024-12-28 04:57:28
28	25	1	154	V_CL_CTA_R	Acid return valve	Closing		2089 2024-12-28 05:02:28
29	26	1	1	V_CL_C	Circulation valve	Opening		2086 2024-12-28 05:02:28
30	25	1	152	V_CL_CTW_S	Fresh water supply valve	Closing		2068 2024-12-28 05:07:28
31	26	1	1	V_CL_C	Circulation valve	Closing		2062 2024-12-28 05:07:28
32	25	1	145	V_CL_B	Bypass valve	Closing		2072 2024-12-28 05:07:28
33	25	1	287	V_CTW_W_S	Fresh water supply	Opening		2151 2024-12-28 08:37:59
34	25	1	160	V_CL_D	Drain valve	Opening		2165 2024-12-28 08:37:59
35	25	1	287	V_CTW_W_S	Fresh water supply	Closing		2121 2024-12-28 08:44:29
36	25	1	156	V_CL_CTC_S	Caustic supply valve	Opening		2067 2024-12-28 08:46:39
37	25	1	160	V_CL_D	Drain valve	Closing		2134 2024-12-28 08:47:39
38	25	1	157	V_CL_CTC_R	Caustic return valve	Opening		2070 2024-12-28 08:47:40
39	25	1	156	V_CL_CTC_S	Caustic supply valve	Closing		2042 2024-12-28 08:51:39
40	25	1	287	V_CTW_W_S	Fresh water supply	Opening		2151 2024-12-28 08:53:21

Slika 4.10. Prikaz podataka iz Excel tablice

je čak 960 otvaranja i 58 operativnih sati, dok je Fresh water supply valve imao isti broj otvaranja, ali nešto manje operativnih sati – njih 45. Bypass valve, iako s manjim brojem operacija (320), ostvario je 53 sata rada.

Podaci iz grafova i tablica otkrivaju zanimljive obrasce u ponašanju ovih ventila. Uočeno je da je vrijeme potrebno za otvaranje u pravilu duže od vremena potrebnog za zatvaranje. Ovi podaci ukazuju na to da mehanički proces otvaranja zahtijeva nešto više vremena nego zatvaranje, što može biti posljedica razlika u opterećenju ili karakteristika sustava tijekom tih operacija.

Dubljom analizom potvrđeno je da broj operacija ima značajniji utjecaj na performanse ventila od ukupnog broja operativnih sati. Drain valve, koji se najčešće koristi, pokazuje i najduže trajanje otvaranja i zatvaranja. S druge strane, Bypass valve, iako ima visok broj radnih sati, ima kraća vremena operacija, što sugerira da veći broj operacija uzrokuje više trošenja mehaničkih dijelova nego sama duljina rada.

\\MATLAB kod za izradu grafova

```
data = readtable('diplomski.xlsx', 'Sheet', 'Query1');
data.LastUpdate = datetime(data.LastUpdate,
'InputFormat', 'yyyy-MM-dd HH:mm:ss');
```



```

openingData145 = data((data.RecNo == 145) &
strcmp(data.TravelingDirection, 'Opening'), :);
closingData145 = data((data.RecNo == 145) &
strcmp(data.TravelingDirection, 'Closing'), :);
openingData160 = data((data.RecNo == 160) &
strcmp(data.TravelingDirection, 'Opening'), :);
closingData160 = data((data.RecNo == 160) &
strcmp(data.TravelingDirection, 'Closing'), :);
openingData287 = data((data.RecNo == 287) &
strcmp(data.TravelingDirection, 'Opening'), :);
closingData287 = data((data.RecNo == 287) &
strcmp(data.TravelingDirection, 'Closing'), :);

figure;
hold on;
plot(openingData145.LastUpdate, openingData145.TravelingTime,
'LineWidth', 2, 'DisplayName', 'Otvaranje');
plot(closingData145.LastUpdate, closingData145.TravelingTime,
'LineWidth', 2, 'DisplayName', 'Zatvaranje');
xlabel('Vrijeme');
ylabel('Vrijeme odziva (ms)');
title('Vrijeme odziva za Bypass valve (V\_CL\_B)');
legend('show');
hold off;

figure;
hold on;
plot(openingData160.LastUpdate, openingData160.TravelingTime,
'LineWidth', 2, 'DisplayName', 'Otvaranje');
plot(closingData160.LastUpdate, closingData160.TravelingTime,
'LineWidth', 2, 'DisplayName', 'Zatvaranje');

```

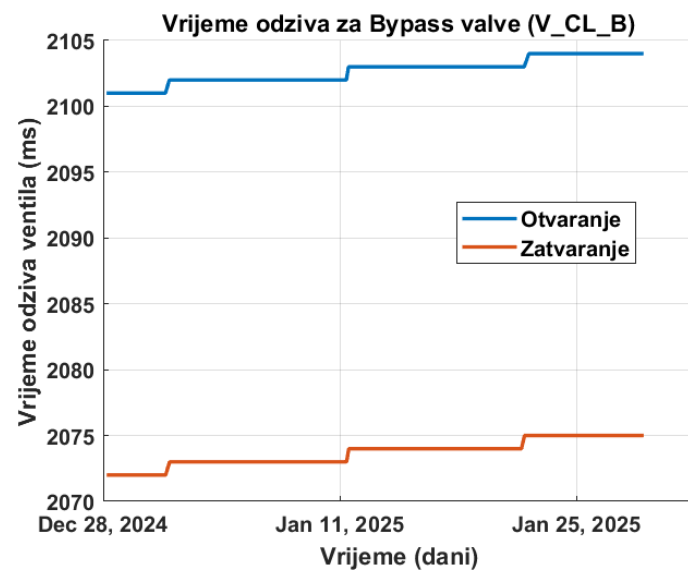
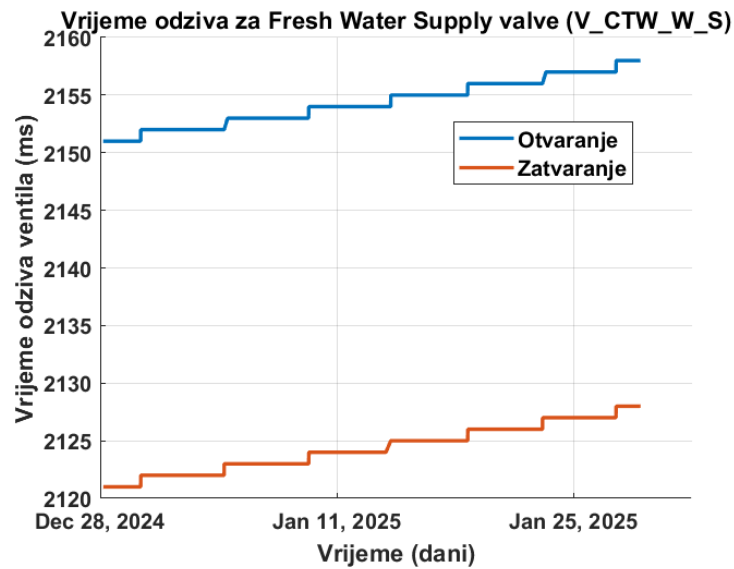
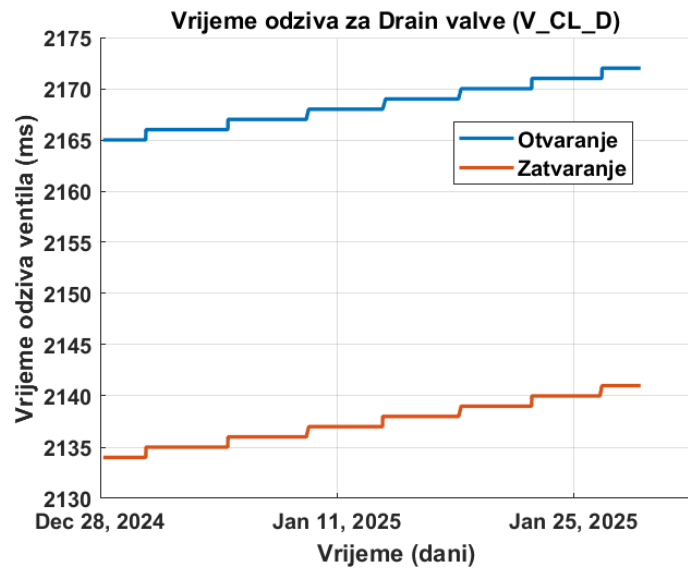
```

xlabel('Vrijeme');
ylabel('Vrijeme odziva (ms)');
title('Vrijeme odziva za Drain valve (V\_CL\_D)');
legend('show');
hold off;

figure;
hold on;
plot(openingData287.LastUpdate, openingData287.TravelingTime,
'LineWidth', 2, 'DisplayName', 'Otvaranje');
plot(closingData287.LastUpdate, closingData287.TravelingTime,
'LineWidth', 2, 'DisplayName', 'Zatvaranje');
xlabel('Vrijeme');
ylabel('Vrijeme odziva (ms)');
title('Vrijeme odziva za Fresh Water Supply valve (V\_CTW\_W\_S)');
legend('show');
hold off;

```

Na slikama 4.11. prikazani su grafovi vremena odziva za pojedine ventile pri otvaranju i zatvaranju. Za generiranje grafova korišten je MATLAB. Iz grafova se vidi da se vrijeme potrebno za otvaranje i zatvaranje ventila postupno povećava. Ovakav trend ukazuje na mogući utjecaj mehaničkog trošenja komponenata ventila, što uzrokuje veće trenje i sporije kretanje mehanizma.



Slika 4.11. Vremena odziva otvaranja i zatvaranja ventila

Rezultati ove analize ukazuju na smjernice koje mogu poboljšati održavanje i optimizaciju rada sustava. Ventili koji imaju visoku učestalost otvaranja i zatvaranja trebali bi se redovito pregledavati kako bi se spriječilo pogoršanje performansi uzrokovano trošenjem mehaničkih komponenti. Premda operativni sati također imaju određeni utjecaj, pokazalo se da broj ciklusa rada ima ključnu ulogu u produljenju vremena potrebnog za operacije. Stoga bi strategija održavanja trebala biti prvenstveno usmjerena na praćenje broja operacija, uz dodatne analize vremena rada. Pratiti trendove vremena otvaranja i zatvaranja kroz dulji period moglo bi pomoći u ranom prepoznavanju potencijalnih mehaničkih problema. Tako bi se omogućilo pravovremeno održavanje te očuvala učinkovitost i pouzdanost sustava.

## **4.2. Predikcija servisa industrijskih ventila**

U industrijskim postrojenjima ključno je osigurati ispravan rad ventila kako bi se održala stabilnost i sigurnost cijelog sustava. Tijekom vremena, uslijed učestalog otvaranja i zatvaranja, dolazi do njihovog trošenja, što se može pratiti kroz parametar vremena odziva, odnosno vremena potrebnog da se ventil potpuno otvori ili zatvori. Prema industrijskim standardima, kada se to vrijeme poveća za 15% posto u odnosu na početnu vrijednost, smatra se da je dostignut prag na kojem je potrebno izvršiti servis ili zamjenu ventila.

Budući da ne postoje točni podaci o početnom vremenu otvaranja ventila, njegova vrijednost procijenjena je na 2000 ms, na temelju informacija dobivenih od stručnjaka s iskustvom u radu s industrijskim ventilima. Na temelju te pretpostavke, prag za servisiranje određen je na 2300 milisekundi. Kako bi se omogućilo pravovremeno održavanje prije nego što dođe do kvara, analizirani su povijesni podaci. Na temelju analize, predviđeno je kada će ventil doseći kritični prag.

Analiza je provedena korištenjem programskog jezika Python, pri čemu je primijenjena linearna regresija kao metoda strojnog učenja za prepoznavanje trenda u podacima. Model je treniran na povijesnim podacima dobivenim iz industrijskog sustava, a korišteni su podaci iz Excel tablice koja je učitana pomoću biblioteke Pandas. Nakon učitavanja, podaci su filtrirani kako bi se izdvojili relevantni zapisi, a vremenske oznake su pretvorene u numerički format, što je omogućilo treniranje modela.

Fokus analize bio je na vremenu otvaranja ventila, budući da je ono veće od vremena zatvaranja. Ova odluka donesena je uzimajući u obzir da se u industrijskim uvjetima ventili često otvaraju pod različitim opterećenjima, što može uzrokovati dulje trajanje tog procesa. S obzirom na to da se vrijeme otvaranja povećava zbog trošenja, smatralo se boljim indikatorom trenutka kada će ventil zahtijevati servis ili zamjenu.

Model linearnom regresijom koristi jednadžbu koja opisuje odnos između vremena i vremena otvaranja ventila. Umjesto predikcije vremena otvaranja za buduće datume, primijenjena je inverzna metoda predviđanja kako bi se odredio datum kada će vrijeme otvaranja doseći prag servisiranja.

Dobiveni rezultati pokazali su različite vremenske intervale u kojima će ventili dostići vrijednost od 2300 milisekundi. Drain valve će taj prag doseći 23. kolovoza 2026. godine, Fresh water supply valve 3. studenog 2026., dok će Bypass valve taj prag dostići tek 26. svibnja 2028. godine. Usporedbom rezultata potvrđeno je da učestalost rada ima ključan utjecaj na brzinu trošenja ventila. Ventili s većim brojem otvaranja i zatvaranja, poput drain valvea i fresh water supply valvea, pokazuju raniji datum servisiranja, dok bypass ventil, koji se koristi rjeđe, ima znatno duži predviđeni rok do servisnog intervala.

Korištenjem prediktivne analize moguće je planirati održavanje, čime se smanjuje rizik od nepredviđenih kvarova i optimiziraju troškovi servisiranja. Umjesto reaktivnog pristupa, gdje bi se ventil mijenjao tek nakon otkazivanja, ovakav način analize omogućuje bolje upravljanje resursima i produljenje vijeka trajanja opreme.

Buduća poboljšanja ovog modela mogu uključivati naprednije metode strojnog učenja, poput polinomske regresije ili analize vremenskih serija, kako bi se postigla veća preciznost predviđanja. Također, dodatni podaci o radnim uvjetima, poput temperature, tlaka i broja ciklusa otvaranja, mogli bi povećati pouzdanost modela. Unatoč jednostavnosti primijenjene metode, korištenjem Pythona omogućena je učinkovita analiza trošenja ventila, čime su stvoreni temelji za daljnju optimizaciju strategije održavanja industrijskih sustava.

```

import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

# ----- Definicija klase za linearnu regresiju ----- #
class LinRegression:
    def __init__(self):
        try:
            from sklearn.linear_model import LinearRegression
        except ImportError:
            import pip
            pip.main(["install", "scikit-learn"])
            from sklearn.linear_model import LinearRegression

        self.regressor = LinearRegression()

    def fit(self, X_train, y_train):
        self.regressor.fit(X_train, y_train)

    def predict(self, X_test):
        return self.regressor.predict(X_test)

def reformat_data_for_regression(df):
    return df["LastUpdate"], df["TravelingTime"]

# ----- GLAVNI KOD ----- #
df = pd.read_excel("diplomski.xlsx", sheet_name="Query1")
(RecNo = 160, TravelingDirection = 'Opening')
filtered_data = df[(df["RecNo"] == 160) &
(df["TravelingDirection"] == "Opening")]

```

```

if filtered_data.empty:
    print("Nema podataka za odabrani ventil.")
else:
    filtered_data["LastUpdate"] = pd.to_datetime
    (filtered_data["LastUpdate"])

    X, y = reformat_data_for_regression(filtered_data)
    X = pd.to_datetime(X).map(pd.Timestamp.toordinal)
    .values.reshape(-1, 1)
    y = y.values.reshape(-1, 1)

    model = LinRegression()
    model.fit(X, y)

    target_time = 2300

    predicted_date_ordinal = (target_time - model.
    regressor.intercept_[0])
    / model.regressor.coef_[0]
    predicted_date = pd.Timestamp.fromordinal
    (int(predicted_date_ordinal[0]))

    print(f"Predviđeni datum kada će vrijeme otvaranja
    dostići {target_time} ms:", predicted_date)

```

```

C:\Users\Antun.Blazic\Desktop\DIPLOMSKI>python lin_regression_test.py
Predviđeni datum kada će vrijeme otvaranja Drain valve-a dostići 2300 ms: 2026-08-23 00:00:00

```

```

C:\Users\Antun.Blazic\Desktop\DIPLOMSKI>python lin_regression_test.py
Predviđeni datum kada će vrijeme otvaranja Fresh Water Supply valve-a dostići 2300 ms: 2026-11-24 00:00:00

```

```

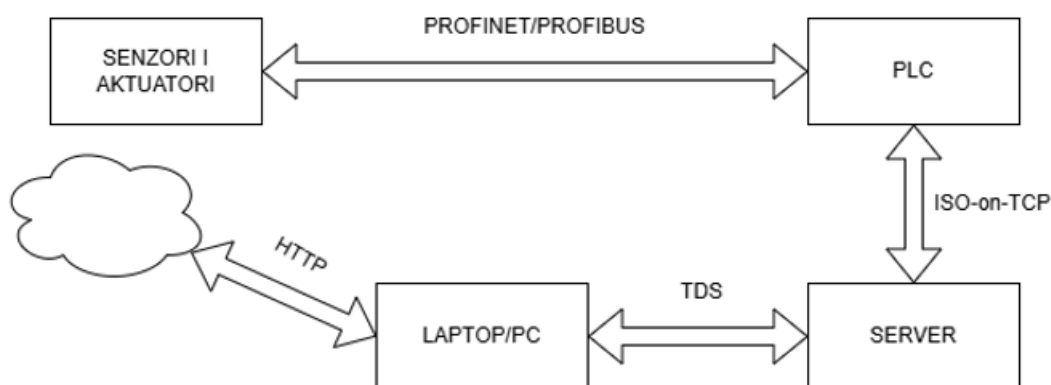
C:\Users\Antun.Blazic\Desktop\DIPLOMSKI>python lin_regression_test.py
Predviđeni datum kada će vrijeme otvaranja Bypass valve-a dostići 2300 ms: 2028-08-02 00:00:00

```

**Slika 4.12.** Prikaz dobivenih predikcija

### 4.3. Cjelokupna arhitektura sustava

Na slici 4.13. prikazana je cjelokupna arhitektura sustava korištena tijekom izrade ovog rada. Uspostava komunikacije senzora i aktuatora s PLC-om ostvarena je putem PROFINET/PROFIBUS komunikacijskog protokola, koji omogućuje brzu i pouzdanu razmjenu podataka u industrijskim sustavima. PROFINET koristi Ethernet tehnologiju za real-time komunikaciju, dok PROFIBUS omogućuje serijsku komunikaciju u automatičkim mrežama. Veza između PLC-a i servera ostvarena je putem ISO-on-TCP (RFC 1006) komunikacijskog protokola. RFC 1006 ("ISO Transport Service on top of the TCP") [9] je proširenje TCP protokola koje omogućuje prijenos ISO transportnih usluga preko TCP/IP mreže dodavanjem dodatnih informacija u zaglavlje poruke. Budući da se temelji na TCP-u (Transmission Control Protocol), koji osigurava pouzdanu i točnu isporuku podataka između dva uređaja putem mehanizama potvrde prijema i ponovnog slanja izgubljenih paketa, RFC 1006 omogućuje prijenos podataka na način orijentiran na poruke, što je ključno za industrijsku automatizaciju i komunikaciju s PLC uređajima. Komunikacija između servera i laptopa ostvarena je korištenjem TDS (Tabular Data Stream) protokola, koji se koristi u SQL Server Management Studio-u za prijenos SQL upita i podataka između klijenta i SQL Servera. Na računalu su se podaci obrađivali i analizirali te su izrađene predikcije koristeći Excel, MATLAB i Python. Za povezivanje s cloudom koristi se HTTP protokol, no u ovom radu umjesto standardnog cloud rješenja korištena je vlastita funkcija istraživačke grupe, prilagođena specifičnim zahtjevima projekta.



Slika 4.13. Prikaz cjelokupne arhitekture sustava



## 5. Zaključak

Tijekom izrade ovog rada analizirani su i implementirani sustavi daljinskog upravljanja dijelom procesa proizvodnje piva, s posebnim naglaskom na automatizaciju sustava čišćenja (CIP – Clean-In-Place). Istraživanjem je utvrđeno da se visoka razina higijene, povećana učinkovitost procesa te smanjenje operativnih troškova mogu postići primjenom naprednih tehnoloških rješenja. Kroz detaljnu analizu arhitekture sustava, upravljačkih komponenti, proceduralnog upravljanja te prikupljanja i obrade podataka, identificirani su ključni aspekti koji doprinose optimizaciji procesa.

Implementacija automatiziranog CIP sustava pokazala se nužnom u modernoj pivarskoj industriji. Preciznom kontrolom parametara čišćenja omogućena je ponovljivost i dosljednost procesa, čime su poboljšani higijenski standardi i osigurana sigurnost proizvoda. Utvrđeno je da automatizacija omogućuje optimizaciju potrošnje vode, kemikalija i energije, čime su postignute ekonomske i ekološke prednosti. Analizom sustava Brewmaxx potvrđeno je da modularna i skalabilna arhitektura pruža fleksibilnost i prilagodljivost različitim veličinama proizvodnih pogona. Integracijom upravljačkih sustava i senzora osigurano je kontinuirano praćenje ključnih parametara procesa, dok je naprednim sustavom alarmiranja i izvještavanja omogućeno pravovremeno reagiranje na moguće probleme, čime su povećani sigurnost i pouzdanost proizvodnog sustava.

Primjenom standarda ISA-88 ustanovljeno je da proceduralno upravljanje doprinosi većoj fleksibilnosti i efikasnosti industrijskih procesa. Jasnim definiranjem procedura i podprocedura omogućeno je precizno izvođenje operacija uz smanjenje mogućnosti pogrešaka. Nadalje, potvrđeno je da se ovim standardom olakšava integracija između različitih industrijskih komponenti te se poboljšava razumijevanje korisničkih zahtjeva.

Jedan od ključnih dijelova ovog istraživanja bio je razvoj sustava za prikupljanje,

obradu i analizu podataka, koji omogućava precizno praćenje rada sustava i pravovremeno donošenje odluka. Implementacijom baze podataka omogućena je sigurna i strukturirana pohrana podataka koji se generiraju tijekom rada CIP sustava. Integracijom senzora i aktuatora omogućeno je kontinuirano praćenje parametara poput temperature, tlaka i koncentracije kemikalija. Podaci su pohranjeni u bazu, čime je omogućena detaljna analiza i optimizacija procesa.

Osim klasičnog praćenja performansi, poseban naglasak stavljen je na prediktivnu analizu podataka, koja omogućava procjenu budućih kvarova i potreba za servisom. Korištenjem algoritama strojnog učenja, poput linearne regresije, analizirani su trendovi u podacima te su predviđeni budući trenuci kada će određene komponente sustava dosegnuti prag servisiranja.

Rezultati su pokazali da se vrijeme otvaranja ventila povećava proporcionalno s brojem radnih sati i brojem ciklusa rada. Na temelju analize utvrđeno je da će Drain valve doseći prag servisiranja 23. kolovoza 2026., Fresh water supply valve 24. studenog 2026., dok će Bypass valve taj prag dostići tek 2. kolovoza 2028. godine. Ovi podaci potvrđuju da ventili koji imaju veći broj ciklusa rada zahtijevaju servis znatno ranije u odnosu na ventile koji rade rjeđe, ali duže vrijeme.

Pravovremenim servisiranjem, temeljenim na prediktivnoj analizi, moguće je značajno smanjiti rizik od nepredviđenih kvarova i zastoja u proizvodnji. Umjesto klasičnog pristupa održavanju, gdje se komponente zamjenjuju tek nakon otkazivanja, ovakav proaktivan pristup omogućava planiranje servisa unaprijed, čime se povećava operativna učinkovitost i smanjuju troškovi popravaka.

## Literatura

- [1] <https://www.proleit.com/brewmaxx-basic-systems>, [mrežno; stranica posjećena: siječanj 2025.].
- [2] <https://www.proleit.com/brewmaxx>, [mrežno; stranica posjećena: siječanj 2025.].
- [3] <https://www.proleit.com/brewmaxx/process-control-systems-for-breweries>, [mrežno; stranica posjećena: siječanj 2025.].
- [4] [https://www.proleit.com/fileadmin/user\\_upload/english/content/04\\_brewmaxx/brewmaxx\\_Brochure\\_brewmaxx\\_System-Brochure\\_EN.pdf](https://www.proleit.com/fileadmin/user_upload/english/content/04_brewmaxx/brewmaxx_Brochure_brewmaxx_System-Brochure_EN.pdf), [mrežno; stranica posjećena: siječanj 2025.].
- [5] <https://www.scribd.com/document/781583067/Brewmaxx-V9-Training-Manual-Automation-SC4U-V1-0>, [mrežno; stranica posjećena: siječanj 2025.].
- [6] <https://www.unitronicsplc.com/what-is-plc-programmable-logic-controller>, [mrežno; stranica posjećena: siječanj 2025.].
- [7] [https://www.plcademy.com/isa-88-s88-batch-control-explained/?utm\\_content=cmp-true](https://www.plcademy.com/isa-88-s88-batch-control-explained/?utm_content=cmp-true), <https://www.plcademy.com/isa-88-s88-batch-control-explained>, [mrežno; stranica posjećena: siječanj 2025.].
- [8] Damir Habuš, “Automatizacija kotlova komine u procesu varionice piva i programiranje funkcija za automatsko generiranje tehničkog opisa”, Diplomski rad, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2015.
- [9] <https://support.industry.siemens.com/cs/document/15048962/what-is-rfc1006-and-what-do-i-need-this-service-for-?dti=0&lc=en-CN>, [mrežno; stranica posjećena: veljača 2025.].

# Sažetak

## Daljinsko upravljanje dijelom procesa proizvodnje piva

Antun Blažić

Ovaj rad bavi se daljinskim upravljanjem dijelom procesa proizvodnje piva, s naglaskom na automatizaciju sustava čišćenja (CIP – Clean-In-Place). Implementacija automatiziranog CIP sustava omogućila je povećanje higijenskih standarda, smanjenje operativnih troškova te optimizaciju potrošnje resursa poput vode, kemikalija i energije. Korištenjem Brewmaxx sustava osigurano je modularno i fleksibilno upravljanje procesima, dok je primjenom standarda ISA-88 poboljšana struktura proceduralnog upravljanja, čime su povećana efikasnost i sljedivost postupaka.

Poseban naglasak stavljen je na prikupljanje i obradu podataka, koji su ključni za praćenje performansi sustava i donošenje informiranih odluka. Implementacijom senzora i aktuatora omogućeno je kontinuirano praćenje radnih parametara, dok je pohrana podataka u bazu osigurala arhivu potrebnu za analizu i optimizaciju procesa. Korištenjem prediktivne analize i algoritama strojnog učenja, poput linearne regresije, analizirani su trendovi trošenja industrijskih ventila. Na temelju prikupljenih podataka predviđeni su trenuci kada će pojedini ventili doseći prag servisiranja, čime je omogućeno pravovremeno održavanje i smanjenje rizika od kvarova.

**Ključne riječi:** automatizacija, PLC, CIP sustav (Clean-In-Place), Brewmaxx, SQL, Python

# Abstract

## Cloud-based control of a brewing production process segment

Antun Blažić

This paper focuses on remote control of a segment of the beer production process, with an emphasis on the automation of the Clean-In-Place (CIP) system. The implementation of an automated CIP system has improved hygiene standards, reduced operational costs, and optimized resource consumption, including water, chemicals, and energy. The use of the Brewmaxx system has enabled modular and flexible process management, while the application of the ISA-88 standard has enhanced procedural control, improving efficiency and traceability.

Special emphasis is placed on data collection and processing, which are crucial for monitoring system performance and making informed decisions. The implementation of sensors and actuators has enabled continuous monitoring of operational parameters, while data storage in a database has provided an archive necessary for analysis and process optimization. By applying predictive analysis and machine learning algorithms, such as linear regression, trends in industrial valve wear have been analyzed. Based on the collected data, the moments when individual valves will reach their servicing threshold have been predicted, enabling timely maintenance and risk reduction of failures. This approach optimizes system performance, increases equipment reliability, and allows for better maintenance planning, reducing costs and ensuring the stability of industrial processes.

**Keywords:** Automation, PLC, CIP system (Clean-In-Place), Brewmaxx, SQL, Python