

Regulacije razine u spremniku pomoću programabilnog logičkog kontrolera

Parmačević, Luka

Master's thesis / Diplomski rad

2021

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:903775>

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

Download date / Datum preuzimanja: **2024-12-25**

Repository / Repozitorij:

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



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

Sveučilišni studij

**REGULACIJA RAZINE U SPREMNIKU POMOĆU
PROGRAMABILNOG LOGIČKOG KONTROLERA**

Diplomski rad

Luka Parmačević

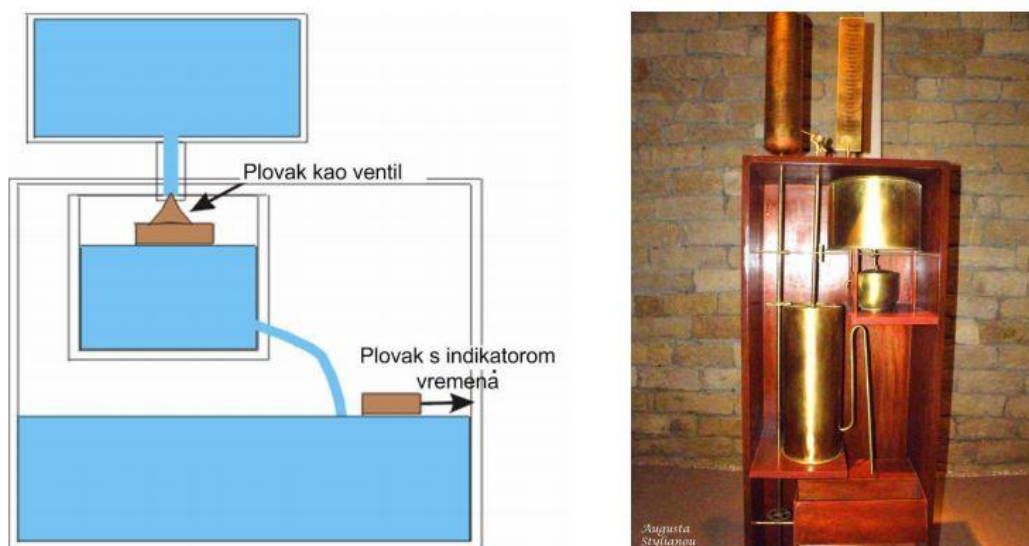
Osijek, 2021.

1. UVOD.....	1
2. REGULACIJSKI KRUG I STANDARDNI TIPOVI REGULATORA	4
2.1. Osnovno o regulacijskom krugu.....	4
2.2. Opis sustava matematičkim modelom.....	5
2.3. Standardni PID regulatori	6
2.4. Podešavanje parametara PID regulatora	9
3. MODELIRANJE PROCESA USKLADIŠTENJA TEKUĆINE U SPREMNIKU S CRPKOM.....	11
3.1. Općenito o MATLAB-u.....	11
3.2. Proces uskladištenja tekućine u spremniku s crpkom.....	11
3.3. Modeliranje procesa uskladištenja tekućine s crpkom.....	12
3.4. Simulacija u simulinku	16
4. OPIS KORIŠTENJA TEHNOLOGIJA ZA SIMULACIJU I VIZUALIZACIJU PROCESA.....	17
4.1. PLC.....	17
4.2. TIA Portal.....	19
4.2.1. PID Compact	24
4.2.2. HMI	27
5. SIMULACIJA I VIZUALIZACIJA PROCESA USKLADIŠTENJA U OKVIRU TIA PORTALA	29
5.1. Implementacija procesa uskladištenja tekućine i regulacije u FBD jeziku	31
5.2. Pokretanje programa u PLC-u i vizualizacija pomoću HMI.....	38
6. ZAKLJUČAK.....	44

1. UVOD

Automatizacija predstavlja tehniku i teoriju koja proces čini automatskim, samogibajućim ili samoupravljanim. Proces automatizacije nastavak je mehanizacije jer proces možemo učiniti automatskim samo ako je dovoljnoj mjeri mehaniziran. Pravilnim razvojem automatizacijske industrije moguće je povećati kvalitetu proizvodnje, proširiti proizvodnju, razviti efikasniji sustav nadzora i upravljanja, međutim cijena svega toga je smanjeni broj ljudskih resursa odnosno radnih mjesta. Prvi uređaji za koje možemo reći da su imali neku vrstu automatizacije datiraju još iz doba Aleksandrije, a izumitelj je bio Ktesibios sa svojim vodenim satom. Vodeni sat možemo vidjeti na slici 1.1 koji se temeljio na regulaciji protoka vode pa je tako mehanizam pokazivao vrijeme trajanja punjenja posude vodom [1].

Prvi automatizirani industrijski proces je bio vodeni mlin za mljevenje brašna i izumljen u 18. stoljeću. Automatsko regulirani mlin prethodnik je velikog izuma centrifugalnog regulatora za regulaciju brzine parnog stroja u drugoj polovici 18. stoljeća. Tim izumom James Watt uvelike je doprinio daljnjem razvoju procesa automatizacije. Princip rada ovog regulatora temeljio se na njihalu koje se u ovisnosti o brzini rotacije spušta ili podiže te na taj način preko poluosovine zatvara ili otvara prigušni ventil. Ventil se nalazio na cijevi za dovod pare, te tako utjecao na brzinu vrtnje stroja. Daljnjim razvojem automatizacije počela se koristiti relejna logika za potrebe elektrifikacije industrijskih tvornica [1].



Sl. 1.1 Vodeni sat iz doba Aleksandrije [1]

Znanstvenici Lagrange i Hamilton u 19. stoljeću pokrenuli su analizu gibanja dinamičkih sustava na temelju diferencijalnih jednadžbi. Britanski znanstvenik Airy osmislio je sustav s povratnom vezom za usmjeravanje teleskopa. Zbog pojave brojnih oscilacija i nestabilnosti, prvi je u povijesti analizirao stabilnost povratne na temelju diferencijalnih jednadžbi.

Razvojem telefonske komunikacije učestali problem koji se javljao unutar komunikacije bio je slab signal te izobličenje signala. Iz tog razloga Harold Black osmislio je pojačalo s negativnom povratnom vezom. Kod takvog sustava pojačala s negativnom povratnom vezom pojačanje je bilo vrlo malo, a sustav je bio otporan na promjene parametara i vanjske smetnje. Tijekom drugog svjetskog rata za potrebe sustava navigacije i kontrole paljbe njemački matematičar Flugge-Lotz izradio je sustav diskontinuiranog automatskog upravljanja.

Početak doba hladnoga rata, razvitak balističkih projektila i svemirske tehnologije temeljio se na razvoju automatske regulacije. Novi sustavi, nove tehnologije i puno bolje analize i sinteze sustava zamijenile su zastarjele teorije zasnovane na frekvencijskom odzivu. Rafinerija tvrtke Texaco 1959. godine bila je prvo postrojenje kojem se upravljalo digitalno, tek nakon toga u industriji se počinju koristiti programabilni logički sklopovi (eng. *Programmable Logical Controller* - PLC). PLC-ovi su bili puno fleksibilniji i pouzdaniji od releja, te su mogli zamijeniti veliki broj releja te su tako bili značajno ekonomski i prostorno isplativiji. Od svog izuma pa do danas PLC se veoma razvio, pa tako danas može izvršavati veliki broj funkcija. Jedan dio problema vezanih uz automatizaciju riješen je pojavom PLC-a, a drugi dio vezan uz centralni nadzor i upravljanje riješen je razvojem SCADA sustava. SCADA je sustav za nadzor, mjerenje i upravljanje industrijskim sustavima [1].

Tema diplomskog rada je izrada upravljačkog sustava za regulaciju tekućine u spremniku pomoću PLC-a. Prema tehničkoj dokumentaciji sustav čini spremnik u koji ulazi voda, crpka, senzor za razinu, senzor za odtok. Unutar programskog paketa MATLAB potrebno je izraditi matematički model, nacrtati odgovarajući blok dijagram te pokrenuti simulaciju u Simulinku. Unutar softvera Siemens TIA Portal potrebno je izraditi cjelokupni projekt odnosno dodati odgovarajuću konfiguraciju, napisati program za regulaciju te izraditi sučelje za komunikaciju između čovjeka i postrojenja. Po završetku provjeriti ispravnosti, pokrenuti simulaciju programa.

U drugom poglavlju obrađeni su osnovni pojmovi vezani uz regulacijski krug i standardne tipove regulatora. Tako je kroz nekoliko potpoglavlja opisan sustav matematičkog modela za proces uskladištenja tekućine iz spremnika pomoću crpke, objašnjeni su standardni tipovi regulatora te podešavanje parametara za PID regulaciju. Modeliranje procesa uskladištenja

tekućine iz spremnika pomoću crpke u programskom paketu MATLAB dano je u trećem poglavlju. Obrađen je programski paket MATLAB, te je unutar dijela MATLAB Simulinka izrađen blokovski dijagram procesa, te je izvršena simulacija procesa i prikazani odzivi. Peto poglavlje daje uvid u izradu programskog rješenja procesa unutar programskog paketa TIA portal, obrađen je programski paket TIA portal i programabilni logički kontroler. Izvršena je simulacije, te su prikazani svi odzivi mjerenja. U petom poglavlju dana je i analiza dobivenih odziva. Na kraju rada dan je zaključak.

2. REGULACIJSKI KRUG I STANDARDNI TIPOVI REGULATORA

2.1. Osnovno o regulacijskom krugu

Automatska regulacija prema definiciji je automatsko održavanje nekog stanja ili mijenjanje tog stanja po određenom zakonu bez obzira na vanjske i unutarnje poremećaje. Isto se ostvaruje uz pomoć povratne veze odnosno usporedbom mjerene i željene veličine. Veličina koja se regulira je mjerena i ista se uspoređuje s željenom vrijednošću (često se naziva referentna veličina), slijedom razlike tih vrijednosti sustav mijenja svoje stanje odnosno, kako bi regulirana veličina dobila zadani iznos. Sustavi automatske regulacije prikazuju se pomoću grafičkih simbola odnosno blok dijagrama te se međusobno povezuju pomoću krugova, strijelica i čvorišta.[1] Primjer blok dijagrama regulacijskog kruga prikazan je na slici 2.1. gdje je:

$r(t)$ - referentna veličina

$e(t)$ - regulacijsko odstupanje

$u(t)$ - postavna veličina

$d(t)$ – poremećajna veličina

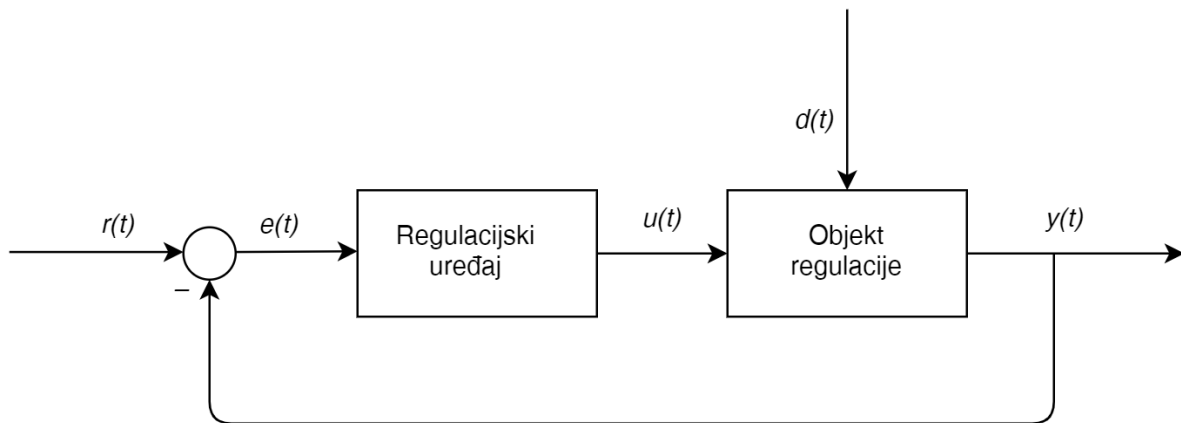
$y(t)$ – regulirana veličina

Prednosti sustava koji sadrži povratnu vezu su:

- otpornost na vanjske poremećaje i promjene vlastitih svojstava,
- proces sastavljen od loših sastavnica može dati dobre rezultate,
- nestabilan sustav može se stabilizirati,
- mogućnost stvaranje poželjnog ponašanja koje nije moguće bez povratne veze.

Osnovni zadatak povratne veze prilikom provedbe regulacije je zajamčiti stabilnost sustava. Stabilnost sustava može se definirati pomoću pobude sustava pa tako ako je pobuda ograničenih vrijednosti, sustav koji je stabilan imat će ograničen odziv. U slučaju nestabilnog sustava odziv bi bio neograničen, te postoji mogućnost pojave oscilacija.

Pojam vođenje sustava obuhvaća regulaciju i upravljanje složenih sustava pomoću računala. Na slici 2.1 prikazan je osnovni oblik blok dijagrama. Osnovna zadaća povratne veze obavlja se pomoću komparatora, a zbog stabilnosti sustava u većini slučajeva nužno je da veza bude negativna. Prilikom uspostave matematičkog modela nužno se pridržavati propisanih pravila za blokove. Uspostava matematičkog modela označava početak matematičke analize nekog sustava.



Sl. 2.1 Osnovni blok dijagram [1]

2.2. Opis sustava matematičkim modelom

Za ispravno vođenje sustava potrebno je dobro poznavanje svih karakteristika i parametara sustava. Poznavanje sustava i njegovih karakteristika omogućava uspostavu matematičkog modela što predstavlja početni korak pri matematičkoj analizi sustava. Razlikujemo dva osnovna pristupa pri izradi modela sustava: pristup po toku energije i pristup po toku signala. Pristup po toku energije temelji se na grafovima, dok se pristup po toku temelji na blok dijagramima. Za izradu matematičkog modela nužno je definirati vezu između ulaza odnosno pobude i izlaza odnosno odziva nekog sustava, ta veza definira se pomoću diferencijalnih jednadžbi. Za jednostavnije rješavanje diferencijalnih jednadžbi uvedena je prijenosna funkcija (eng. *transfer function*), koja se uz pomoć Laplaceove transformacije koristi u daljnjoj matematičkoj analizi linearnih sustava. Možemo reći da definirati matematički model znači napisati jednu ili više diferencijalnih jednadžbi sustava. Za potpunu analizu vođenja nekog sustava potrebno je sustav pomoću softvera simulirati na računalu. Rezultati simulacije prikazuju se grafički, najčešće putem vremenskog tijeka procesnih veličina od interesa.

Matematički model moguće je dobiti teorijskim ili analitičkim putem. Teorijski način se temelji na zakonu o održavanju mase, energije i drugim zakonima fizike. Model napisan pomoću diferencijalnih jednadžbi opisuje promjene stanja stvarnog tehničkog sustava tijekom vremena kao posljedicu djelovanja vanjskih pobuda ili unutarnje pohranjene energije. Analitički način dobivanja modela je u prednosti radi lakšeg praćenja ponašanja sustava. Prijenosna funkcija predstavlja poveznicu između ulaza i izlaza nekog sustava, odnosno uzrok i posljedicu promjena. Modelirani sustavi najčešće se prikazuju pomoću blok dijagrama, prilikom sastavljanja modela potrebno je poštovati pravila operacija između blokova. Strukturni blok dijagrami predstavlja prijenosnu funkciju i odgovarajuće označeni tok signala. Bitno svojstvo strukturnih dijagrama je mogućnost proširenja blokova, pa tako od jednostavnih nastaju složeni blok dijagrami. Funkcionalni blok dijagrami prikazuju funkcionalne odnose između elementa, za crtanje takvih blok dijagrama nije potrebno čvrsto držati se matematičkih pravila. Većina sustava u stvarnosti je nelinearna ili je barem jedan element regulacijskog kruga nelinearan, samim time je i sam sustav nelinearan. Da bismo mogli pristupiti analizi i sintezi nelinearnih sustava potrebno je sustav linearizirati.

Pod pojmom linearizacije podrazumijeva se aproksimacija nelinearnog matematičkog modela procesa linearnim, u okolišu radne točke. Radna točka predstavlja jedno od stacionarnih stanja. Linearizacija se provodi razvojem funkcije u Taylorov red u okolni radne točke, te odbacivanjem članova čiji je red viši od 2. Prije formiranja modela možemo zanemariti određene nelinearne odnose, s tim da je vrlo bitna dobra procjena kako se zanemarivanjem ne bi izgubilo na preciznosti rješenja sustava.

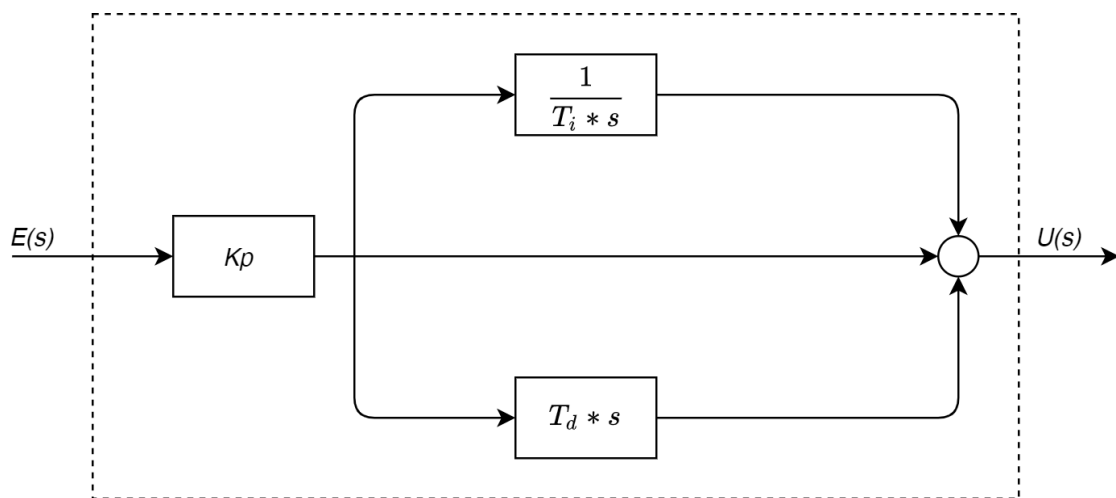
2.3. Standardni PID regulatori

Regulator tvori regulacijsko odstupanje $e(t) = x_r(t) - y(t)$, koje se dalje obrađuje kako bi dobili upravljačku veličinu $u(t)$ koja pomoću izvršnog člana održava kontrolirani tok energije procesa. Na taj način regulirana veličina se održava na zadanom iznosu i tijekom djelovanja poremećaja. Promjena upravljačke veličine $u(t)$ ne događa se istog trenutka zbog vremenskog zatezanja odnosno nemogućnosti da se energetske spremnici trenutačno pune ili prazne. Brzina promjena stanja se temeljni na vremenskim konstantama, a dok struktura i parametri regulatora proizlaze iz matematičkog modela procesa i zahtjeva na zatvoreni regulacijski krug. Najčešće se koriste linearni regulatori koji se mogu sastojati od tri djelovanja. Radi se o proporcionalnom (P), integralnom (I) i derivacijskom (D) regulatoru. Regulator I i D rijetko se nalaze zajedno, a u kombinaciji s P regulatorom tvore proporcionalno-integralno-derivacijski (PID) regulator.

Kombinacije P i I regulatora odnosno P i D regulatora tvore proporcionalno-integralni (PI) odnosno proporcionalno-derivacijski (PD) regulator [2].

PID regulator (vidi sliku 2.2.) jedan je od najčešće korištenih regulatora u automatizaciji, regulator se sastoji od 3 kombinirana regulacijska djelovanja. Njegova glavna prednost je kvaliteta odnosno brzina i točnost odziva, te zadovoljavanje maksimalnih granica odstupanja.

Odabirom odgovarajućih parametara regulatora, vladanje procesa se podešava pomoću regulatora tako da se postigne najbolje regulacijsko vladanje. PID regulator čini zbroj proporcionalnog, integralnog i derivacijskog djelovanja kao što je dano izrazom (2-1).



Sl. 2.2. Blokovski prikaz PID regulatora [2]

$K_R = K_p$ - koeficijent pojačanja (engl. *Proportional Gain*)

$T_I = \frac{K_p}{K_I}$ - integralna vremenska konstanta (engl. *Integration Time*)

$T_D = \frac{K_D}{K_p}$ - derivacijska vremenska konstanta (engl. *Derivative Time*)

K_R, T_I, T_D - parametri regulatora koji se mogu podešavati.

Određivanje parametara regulatora naziva se sinteza regulatora.

$$u(t) = K_p e(t) + \frac{K_R}{T_I} \int_{t_0}^t e(t) dt + K_R T_D \frac{de(t)}{dt} \quad (2-1)$$

Djelovanje regulatora najjednostavnije se može objasniti razdvajanjem na pojedina regulacijska djelovanja. P djelovanje prikazano je jednadžbom (2-2):

$$u_p = K_p \cdot e(t) \quad (2-2)$$

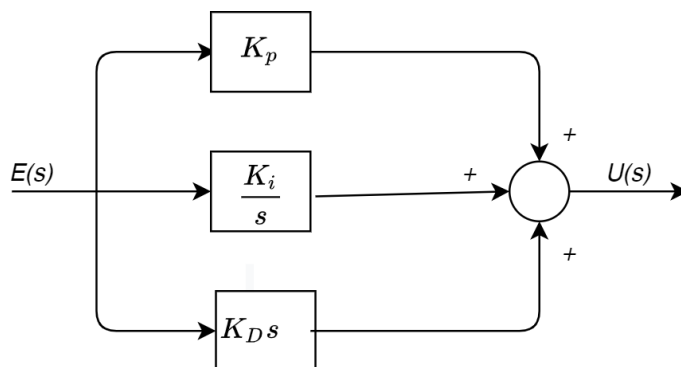
gdje K_p označava proporcionalno pojačanje regulatora. P član je proporcionalan s iznosom regulacijskog odstupanja te se toga trajno regulacijsko odstupanje se ne može otkloniti čistim P djelovanjem. Povećanjem K_p pojačanja vrijeme porasta se smanjuje do zadane vrijednosti, a povećavaju se oscilacije sustava. I djelovanje prikazano je jednadžbom (2-3):

$$u_i(t) = \frac{K_p}{T_i} \int_{t_0}^t e(t) dt \quad (2-3)$$

gdje K_i predstavlja integralno pojačanje regulatora, jednako je omjeru $\frac{K_p}{T_i}$. Djelovanje integralnog člana ima djelovanje proporcionalno vremenskom integralu regulacijskog odstupanja i njegovoj veličini, time se u sustav unosi kašnjenje jer dolazi do akumuliranja prethodnih stanja, ali isto tako unosi se poboljšanje jer se uklanja regulacijsko odstupanje u stacionarnom stanju i smanjuje vrijeme porasta do zadane vrijednosti. D djelovanje prikazano je jednadžbom (2-4):

$$u_D(t) = K_p T_d \frac{de(t)}{dt} \quad (2-4)$$

gdje K_D je derivacijsko pojačanje regulatora, jednako je umnošku $K_p T_d$. Derivacijski član djeluje prediktivno na iznos regulacijskog odstupanja te se zato smanjuje vrijeme smirivanja oscilacija i povećava stabilnost sustav. Na slici 2.3. prikazana je blok shema idealnog PID regulatora u paralelnoj izvedbi.



Sl. 2.3. Blok shema idealnog PID regulatora – paralelna [1]

Osim što se može formirati paralelna struktura moguće su serijska struktura te neke verzije za određene slučajeve, kao što su I-PD ili PI-D strukture. Prijenosna funkcija idealnog PID regulatora prikazana je jednačinom (2-5).

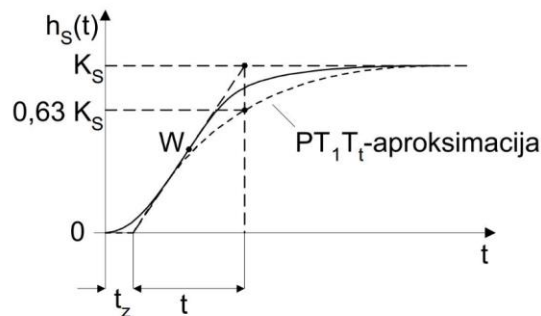
$$G_R = \frac{U(s)}{E(s)} = K_p + \frac{K_I}{s} + K_D s = K_R \left(1 + \frac{1}{T_I s} + T_D s\right) \quad (2-5)$$

2.4. Podešavanje parametara PID regulatora

Postoji nekoliko mogućnosti kako se može formirati regulator s PID djelovanjem. Parametre regulatora moguće je odrediti na temelju matematičkog modela procesa i zahtjeva koji se postavljaju na zatvoreni regulacijski krug. Međutim, mnogi procesi su izrazito složeni pa matematički model procesa nije dostupan. Stoga se parametri regulatora određuju eksperimentalno. Neke od eksperimentalnih metoda kojima je moguće podešavati parametre PID regulatora su metoda Ziegler–Nichols, Cohen–Coon metoda, podešavanje po IMC postupku i druge. Ziegler-Nichols metoda se temelji na metodi prijelazne funkcije otvorenog regulacijskog kruga i metodi ruba stabilnosti u zatvorenom regulacijskom krugu [3].

Metoda prijelazne funkcije se upotrebljava u situacijama gdje regulirani sustav nije moguće dovesti do ruba stabilnosti, te se bazira na odzivu sustava na skokovitu promjenu ulazne veličine. Da bi se ostvarilo isto potrebno je regulator postaviti u ručni rad i u trenutku kada se sustav ustali izazvati skokovitu promjenu veličine koja se regulira. Iz dobivenog grafa (vidi sliku 2.4.) prijelazne funkcije $h_s(t)$ na temelju tangente potrebno je očitati u točki infleksije vrijednosti

K_s / t_a i vrijeme zadržavanja t_z . Prema podacima iz tablice 2.1 se mogu dobiti potrebne vrijednosti PID parametra [3].



Sl. 2.4. Odziv sustava na skokovitu promjenu referentne veličine – Ziegler – Nichols metoda prijelazne funkcije[3]

Metoda ruba stabilnosti koristi se na način da se unutar regulacijskog kruga isključe I i D djelovanja PID regulatora. Pojačanje K_r regulatora potrebno je povećavati sve dok se ne dobije trajne oscilacije izlazne veličine. Pojačanje uz koje se dobiju trajne oscilacije naziva se kritično pojačanje K_{rkr} . Na temelju pojačanja K_{rkr} i periode T_{kr} trajnih oscilacija definiraju se vrijednosti parametara regulatora prema izrazima iz tablice 2.2.

Tablica 2.1. Tablica za određivanje parametara putem Ziegler-Nichols metoda rub stabilnosti[3]

Regulator	K_R		T_I	T_D
P	$0,5 K_{rkr}$		-	
PI	$0,45 K_{rkr}$		$0,85 K_{rkr}$	
PID	$0,6 K_{rkr}$		$0,5 K_{rkr}$	$0,12 K_{rkr}$

Tablica 2.2. Tablica za određivanje parametara putem Ziegler-Nichols metoda prijelazne funkcije [3]

Regulator	K_R	T_I	T_D
P	$1 / K_s * t_a / t_z$	-	-
PI	$0,9 / K_s * t_a / t_z$	$3,33 t_z$	-
PID	$1,2 / K_s * t_a / t_z$	$2 t_z$	$0,5 t_z$

3. MODELIRANJE PROCESA USKLADIŠTENJA TEKUĆINE U SPREMNIKU S CRPKOM

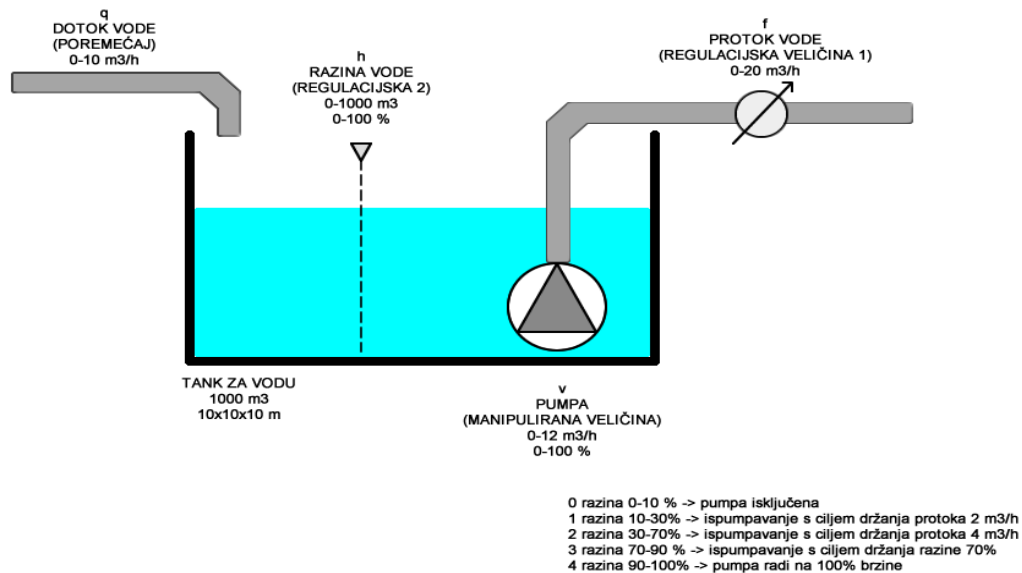
3.1. Općenito o MATLAB-u

MATLAB je programski paket koji omogućava jednostavnije, efikasnije i brže rješavanje složenih matematičkih zadataka. Paket omogućava simulacije, matrično računanje, programiranje i mnoge druge značajke vezane uz regulaciju, upravljanje te vizualizaciju. Prva verzija programa primjenjivala se 1970. godine na sveučilištima University of New Mexico i Stanford University gdje se koristio u numeričkoj analizi, linearnoj algebri i računskim operacijama. MATLAB je danas znatno napredniji nego u to vrijeme, pa tako danas uz osnovni paket postoji veliki broj proširenja koja pokrivaju praktički sva područja tehnike. Unutar programa moguće je obrađivati signale, 2D i 3D grafičke prikaze, regulaciju, analize u vremenskoj i frekvencijskoj domeni automatsko upravljanje i brojne druge stvari. Korisnik uz određena znanja može izraditi vlastite alate, skripte i biblioteke te modificirati postojeće, programski jezici koje prepoznaje su C, Fortran, Ada ili Java. Jedan od važnih dijelova programskog paketa MATLAB je SIMULINK. To je dio programa u koji omogućava simulacije i vizualizacije sustava. Sastoji se od radnog okruženja u kojem se nalazi prozor za izradu željenog modela i blokova koji se dodaju u model [10].

3.2. Proces uskladištenja tekućine u spremniku s crpkom

Iz analize tehničke dokumentacije, zahtjeva i potrebne konfiguracije ustanovljeno je da postrojenje obuhvaća spremnik (tank za vodu) poprečnog presjeka spremnika $A=1000\text{ m}^3$, crpka za izvlačenje vode iz spremnika te senzor za mjerenje razine vode. Punjenje spremnika s vodom u simulatoru je zadano kao nasumično. U slučaju kada je razina vode u spremniku ispod 10%, crpka ne smije biti uključena. Crpka se uključuje u trenutku kada je razina vode dođe do 10% razine, u intervalu od 10% do 30% crpka ispumpava vodu s ciljem držanja protoka $2\text{ m}^3/h$. U trenutku kada je razina vode između 30% i 70% crpka ispumpava vodu s ciljem držanja protoka $4\text{ m}^3/h$. Kad se razina vode popne između 70% i 90% crpka ispumpava vodu tako da održi razinu vode na 70%. U posljednjem slučaju kada se razina vode iznad 90% crpka radi na 100% brzine. Slika 3.1 prikazuje pojednostavljeni prikaz procesa.

3.3. Modeliranje procesa uskladištenja tekućine u spremniku s crpkom



Sl. 3.1. Pojednostavljeni prikaz procesa uskladištenja tekućine spremniku pomoću crpke.

Gibanje fluida posljedica je djelovanja sile, gibanje fluida unutar cijevi posljedica je tlaka koji se stvara zbog djelovanja sile koju uzrokuje crpka. Količina fluida u spremniku može se prikazati kao funkcija dotoka i odtoka. Osnovna jednadžba ravnoteže mase je [3]:

$$m_u(t) - m_i(t) = \frac{d}{dt} m(t) \quad (3-1)$$

$$m(t) = \rho(t) \cdot V(t) \quad (3-2)$$

$$q_u(t) - q_i(t) = \frac{d}{dt} V(t) \quad (3-3)$$

$$V(t) = A \cdot h(t) \quad (3-4)$$

gdje je:

m_u, m_i - maseni protok fluida [kg/s];

V - volumen uskladištenog fluida [m^3];

ρ - gustoća fluida [kg / m^3];

q_u - ulazna količina fluida (dotok) [m^3 / s];

q_i - izlazna količina fluida (odtok) [m^3 / s];

m - uskladištena masa fluida [kg];

Masa fluida uskladištenog unutar spremnika konstantnog presjeka može se izračunati izrazom:

$$m(t) = A \cdot h(t) \cdot \rho \quad (3-5)$$

gdje je:

A – površina presjeka spremnika [m^2];

h – visina fluida [m];

Ako se pretpostavi da je gustoća fluida konstantna, onda se proces uskladištenja tekućine u spremniku s crpkom definiran je na temelju jednadžbi stanja (3-6),(3-7),(3-8)

$$\frac{dV}{dt} = q_u(t) - q_i(t) \quad (3-6)$$

$$\frac{d(Ah(t))}{dt} = q_u(t) - q_i(t) \quad (3-7)$$

$$A \frac{dh(t)}{dt} = q_u(t) - q_i(t) \rightarrow \frac{dh}{dt} = \frac{1}{A} (q_u(t) - q_i(t)) \quad (3-8)$$

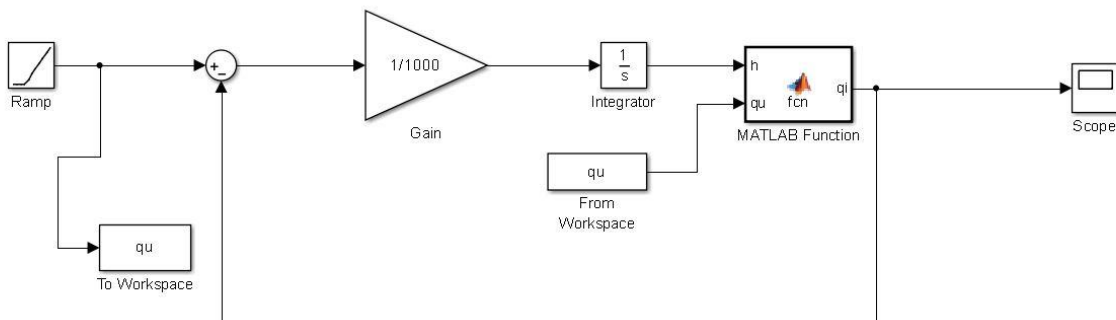
$$q_i(t) = \begin{cases} 0, \text{ ako je } 0 \leq h(t) \leq 10\% \\ 2m^3 / h, \text{ ako je } 10\% \leq h(t) \leq 30\% \\ 4m^3 / h, \text{ ako je } 30\% \leq h(t) \leq 70\% \\ q_i(t), \text{ ako je } 70\% \leq h(t) \leq 90\% \rightarrow q_i(t) = \frac{1}{A} (q_u(t) - (4 + (h(t) - 70\% h_{\max})) * P \\ 12m^3 / h, \text{ ako je } 90\% < h(t) \leq 100\% \end{cases} \quad (3-9)$$

U slučaju kada je razina vode u spremniku između 70% i 90% spremnika primjenjuje se regulacija s ciljem održavanja razine na 70%. Iz tog razlog vrijedi

$$q_i(t) = \frac{1}{A} (q_u(t) - (4 + (h(t) - 70\% h_{\max})) * P) \quad (3-10)$$

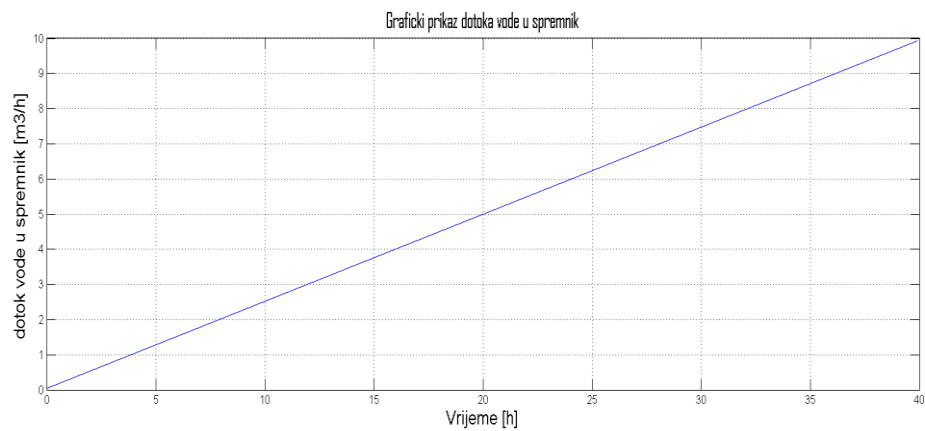
gdje je h_{\max} maksimalna visina fluida u spremnika. Varijabla P predstavlja proporcionalni regulator čija vrijednost uzeta prema simulaciji unutar TIA Portala te ista iznosi 5.

Na slici 3.2. možemo vidjeti blok dijagram za ovaj proces. Dijagram se temelji na matematičkom modelu i jednačbama (3-9) koje opisuju uskladištenje tekućine iz spremnika pomoću crpke promjenjive brzine vrtnje. Model nacrtan je pomoću blokova unutar Simulinka te je pomoću istog izvršena simulacija. Za potrebu simulaciju izlaznog toka izrađena je vlastita m-funkcija. Slika 3.4. prikazuje tu funkciju na kojoj se može vidjeti programski kod za regulaciju ovog procesa.



Sl. 3.2. Prikaz blokovskog dijagrama unutar MATLAB Simulinka

Na slici 3.3. možemo vidjeti grafički prikaz dotoka tekućine u spremnik. Pobuda je u Simulinku modelirana pomoću bloka *ramp* te se pomoću njega simulira dotok tekućine u spremniku.



Sl. 3.3. Grafički prikaz dotoka vode u spremnik

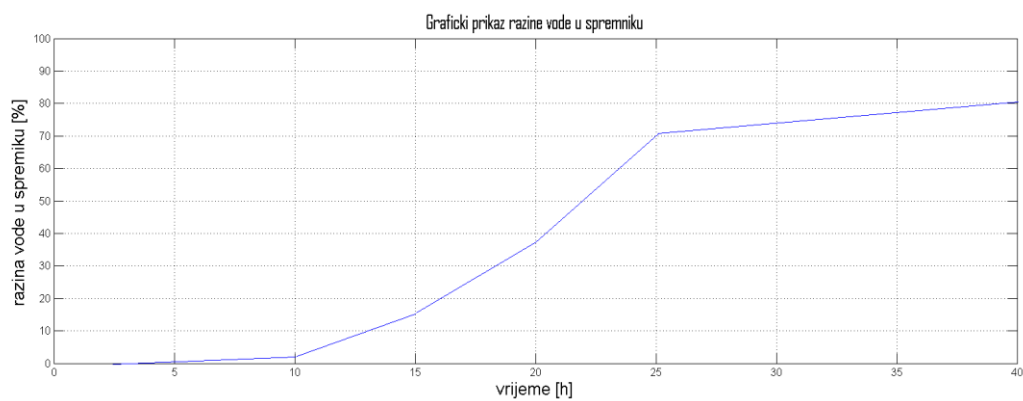
```
MATLAB Function* x
1 function qi= fcn(h, qu)
2     %#codegen
3     if(h>0 && h<=10)
4         y=0;
5     elseif (h>10 && h<=30)
6         y=2;
7     elseif(h>30 && h<=70)
8         y=4;
9     elseif(h>70 && h<=90)
10        y=(4+(h-0.7*Hmax))*P;
11    else
12        y=12;
13    end
14
15    qi= qu-y;
```

Sl. 3.4. MATLAB function unutar Simulinka

3.4. Simulacija u simulinku

Prije pokretanja simulaciji postavljeni su parametri simulacije pa tako trajanje simulacije iznosi 40 h, korišten je *Variable-step (ode45)* solver. Vrijednosti *max* i *initial step size* postavljeni su na auto. *Relative tolerance* iznosi $1e-3$, dok je *absolute tolerance* postavljena na auto.

Sljedeće slike prikazuju nam odzive dobivene simulacijom blokovskog dijagrama unutar Simulinka. Slika 3.5. prikazuje odziv razine vode u spremniku tijekom simulacije. Iz odziva se mogu vidjeti promjene u razini tijekom simulacije.



Sl. 3.5. Grafički prikaz razine vode u spremniku

Na slici 3.6. može se vidjeti grafički prikaz odtoka vode iz spremnika.



Sl. 3.6. Grafički prikaz odtoka vode u spremniku

4. OPIS KORIŠTENJA TEHNOLOGIJA ZA SIMULACIJU I VIZUALIZACIJU PROCESA

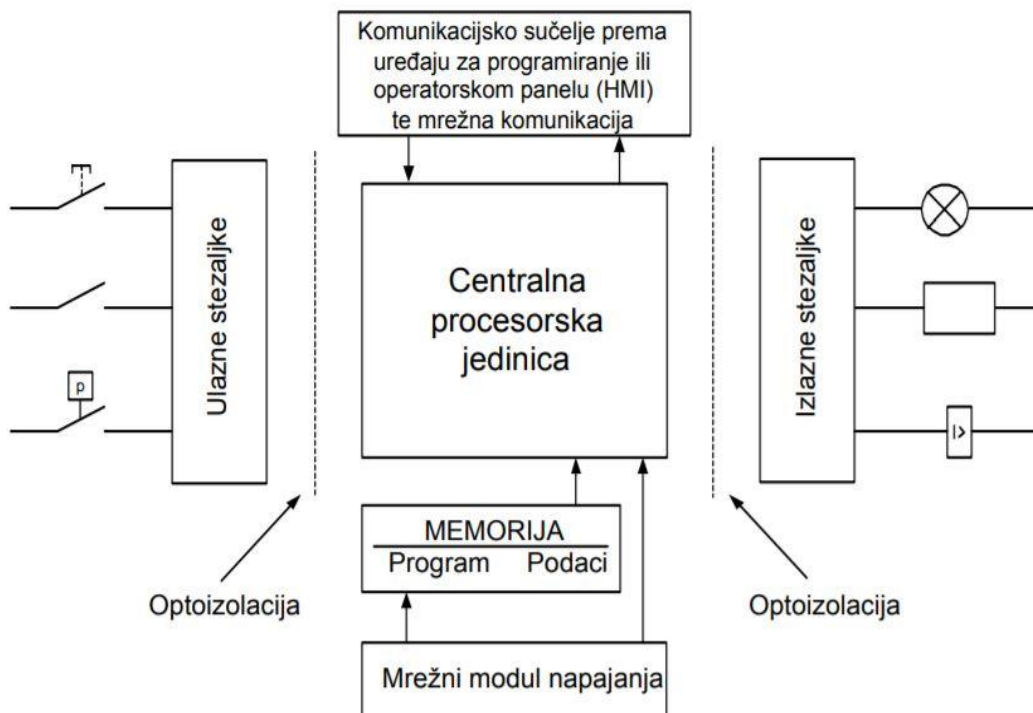
4.1. PLC

PLC (eng. *Programmable Logic Controller*) je programabilni logički kontroler, odnosno elektronički uređaj za automatizaciju s funkcijama upravljanja spremljenih u obliku programa u memoriji upravljačke jedinice. Njegovi ulazi su tipkala, sklopke, pretvornici i/ili senzori. PLC se najviše koristi kao osnovni dio upravljačkih automatskih sustava u industriji. Njegov program, odnosno algoritam, se može jednostavno mijenjati te je pogodan za brza rješenja i aplikacije. Projektiran je za teške uvjete rada, otporan na vibracije, temperaturne promjene, električne smetnje, udarce. Sastoji se od upravljačkog dijela, s jezgrom koju čini procesor s aritmetičkom jedinicom i memorijom te ulazno/izlaznih jedinica. Memoriju čine brojila, zastavice, podaci sustava, mjerači vremena, stanje procesnih ulaza i izlaza te korisnički program. Dio memorije je nepromjenjivog sadržaja (ROM) i u njemu je pohranjen operacijski sustav. Operacijski sustav čine programi koji određuju na koji način će se izvoditi korisnički program, na koji način će se upravljati ulazima i izlazima, memorijom i upravljanjem podataka. Program koji napiše korisnik sprema se u podatkovnu memoriju, RAM memoriju PLC-a. Upravljačka jedinica PLC-a sadrži utor za povezivanje s monitorom pomoću Profibus/MPI sučelja ili etherneteta. Za izvođenje posebnih funkcija PLC se može proširiti dodavanjem odgovarajućih modula. Može sadržavati module za komunikaciju, za digitalne ulaze i izlaze, analogne ulaze i izlaze, module za napajanje, procesorske, funkcijske i povezne module. Na slici 4.1. prikazan je Siemens Simatic S7-1200 model PLC-a [6].



Sl. 4.1. Prikaz PLC-a SIEMENS SIMATIC S7-1200 [9]

Na ulaznom dijelu PLC-a nalaze se stezaljke na koje stižu signali iz upravljanog procesa, to je mjesto prilagodbe dolaznog vanjskog signala kojeg prima i obrađuje procesorska jedinica. Podaci koje PLC prima na svojim ulazima mogu biti digitalne ili analogne. Digitalne (diskretne) informacije koje dolaze na ulaz i izlaz mogu biti signali sa sklopki, tipkala, senzora i sl. Analogni ulazi i izlazi su kontinuirani, varijabilni signali, njihove vrijednosti variraju od 0 do 24 mA odnosno 0 do 10V. Zbog prilagodbe signala uvodi se optička izolacija koja pomaže pri galvanskom odvajanju strujnih krugova koje se uvodi zbog potencijalnih razlika, filtriranja signala i ostalih smetnji. Na digitalni izlaz u obično budu povezani različiti sklopnici, releji, svici, razvodnici, sklopke, signalne lampe i slično. Zadaća tih izlaza je izvršavanje određenih radnji kao što su izdavanje upozorenja, uklapanje releja, uključivanje svjetla, grijača i slično. Na analogni izlaz mogu se povezati strujni signali za pregled veličina na instrumentu, referenca brzine za frekvenciji pretvarač, PID regulirana veličina itd. Stezaljke na izlazu su optički izolirane radi galvanskog odvajanja električnih krugova. Na slici 4.2. možemo vidjeti načelnu shemu rada PLC-a. Na slici 4.2. prikazana je načelna shema rada PLC-a. [4].



Sl. 4.2. Načelna shema rada PLC-a [4]

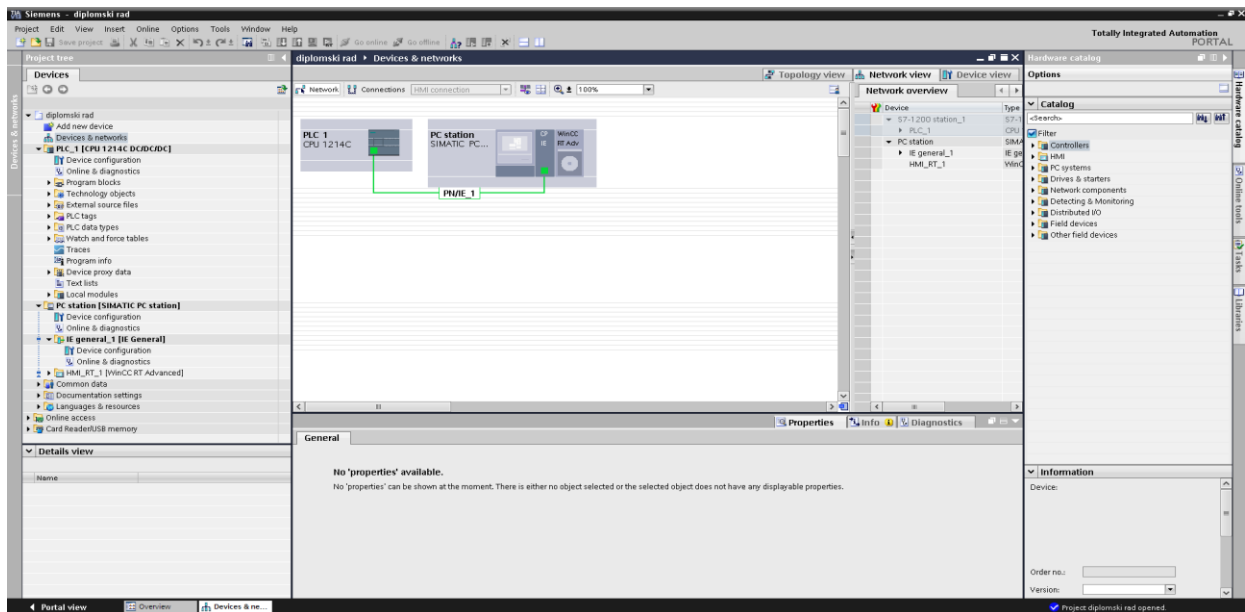
Što se tiče samog rada PLC-a njegov program se izvršava ciklično. Ako dođe do promjene stanja na ulazima PLC mora sustavno ispraviti stanje izlaza onako kako je to zadano logikom programa. Ciklus započinje pokretanjem odnosno startom programa, zatim dolazi do obrade i provjere stanja ulaza. Poslije toga vrši se obrada programa odnosno povezuje se ulazna stanja sa programom (logikom) koji je napisao korisnik te se rezultati šalju na izlaz. Nakon toga dolazi do prijenosa odrađenih informacija na fizičke izlaze PLC-a. Posljednja faza su operacije potrebne za rad operacijskog sustava PLC uređaja i komunikacija sa vanjskim jedinicama. Najpoznatiji proizvođači PLC-ova u svijetu su Siemensov SIMATIC, ABB, Schneider , Mitchubishi , General Electrics i mnogi drugi.

4.2. TIA Portal

TIA portal (eng. *Totally Integrated Automation Portal*) Siemensov je programski paket koji služi za konfiguriranje, programiranje i praćenje rada PLC-a. Program je na tržištu od 2010. godine s ciljem unaprjeđenja automatizacije pomoću integriranih sklopova odnosno projektiranje digitalne automatizacije. Sadrži okvir za optimiziranje plana, procesne i strojne postupke te nam pruža mnoštvo mogućnosti poput integriranje upravljača (kontrolera), izradu HMI (eng. *Human-Machine Interface*), diskove, kontrolu pokreta, upravljanje motorom i ostalo. Prednost korištenja TIA portala je smanjenje troškova, fleksibilnost te postizanje najbolji mogućih rezultata, samo vrijeme projektiranja bitno je smanjeno zbog kvalitete programiranja koda. Softver se sastoji od osnovnog dijela SIMATIC STEP 7, a ostali dijelovi koje softver može sadržavati su SIMATIC WinCC za vizualizaciju, SINAMICS Startdrive za parametiranje pogona, programa SIMCODE koji se koristi za upravljanje motorom. Program pruža korisniku vrlo jednostavno rješavanje zadaća automatskog upravljanja i brzu upotrebu, jednostavno sučelje programa dizajnirano kako bi rad u programu bio što prihvatljiviji početnicima (vidi sliku 4.3). Svi podaci nalaze se u jednom projektu pa je tako lako konfigurirati PLC, napisati kod i izraditi vizualizaciju.

Za izradu rješenja automatizacije potreban je:

- PLC koji upravlja procesom uz pomoć programa, i
- HMI uređaj koji pomaže pri upravljanju i vizualizaciji



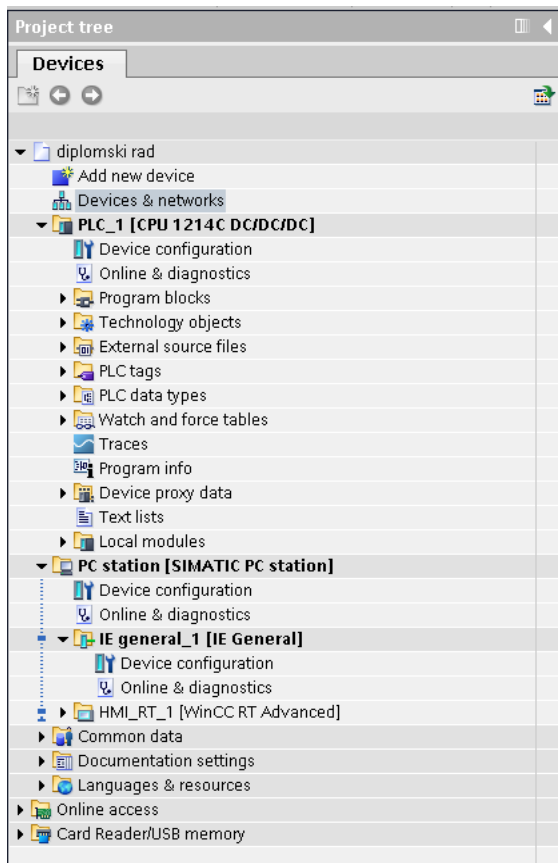
Sl. 4.3. Prikaz sučelja TIA Portala

Prilikom programiranja PLC-a koriste se blokovi:

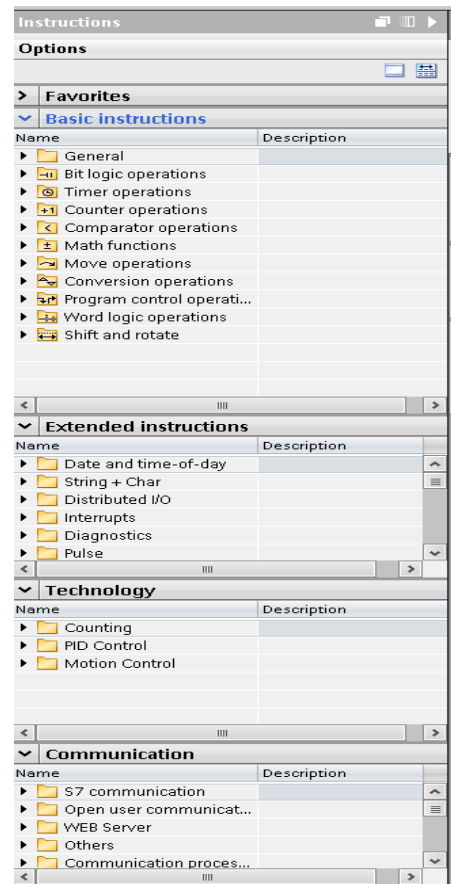
- Organizacijski blok (OB) - struktura programa odnosno glavni glavni blok programa. Služi kao sučelje između operacijskog sustava i korisničkog programa. Može se izvoditi periodički, ovisno o veličini programa,
- Funkcija (FC) - funkcije bez memorije koje sadrže program koji će se izvršavati prilikom poziva funkcije,
- Funkcijski blok (FB) - blok koji ima pridružen podatkovni blok u kojeg sprema vrijednosti unutarnjih parametara pri pozivu statičke varijable bloka,
- Podatkovni blok (DB) – definirana područja radne memorije za spremanje podataka.

WinCC unutar TIA portala služi za izradi vizualizacije odnosno HMI rješenja, vrlo jednostavno i funkcionalno može se prikazati bilo koje postrojenje. WinCC sučelje pruža jednostavnu kontrolu koda, upravljanje ekranima, ubacivanje novih simbola, dohvaćanje potrebnih simbola te nudi gotova rješenja [5].

U prozoru *Project Tree* unutar kartice *Devices* (vidi sliku 4.4.) nalazi se konfiguracija, veza između komponenata odnosno uređaja putem blokova i tagova. Unutar TIA portala moguće je pisati pomoću više programskih jezika: STL (*Statement List*), FBD (*Function Block Diagram*), LAD (*Ladder Diagram*) i SCL (*Structured Control Language*).



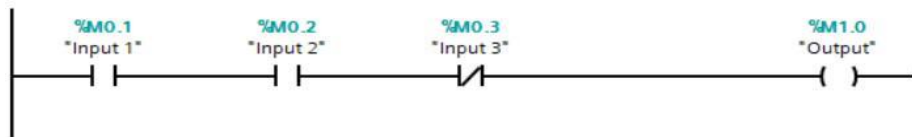
Sl. 4.4. Prikaz prozora *Devices* unutar TIA Portala



Sl. 4.5. Prikaz prozora *Options* unutar TIA Portala

Na slici 4.5. prikazan je prozor *Options*, u njemu se nalazi veliki broj izbornika. Izbornik *Favorites* sadrži funkcije koje korisnik koristi najčešće, nalaze se ovdje kako bi korisniku prilikom pisanja koda bile što dostupnije. U izborniku *Basic Instructions* nalaze se osnovni blokovi koji služe za pisanje programa, dok se blokovi za naprednije instrukcije nalaze u izborniku *Extended instructions*, njihova zadaća je skratiti vrijeme pisanja koda. Izbornik *Technology* sadrži gotove brojače, regulatore i kontrolu gibanja i itd.

LAD kontaktni dijagrami nastali su na temelju električnih upravljačkih shema relejne logike koju je trebao zamijeniti PLC. Sheme su prikazivale redosljed paljenja i gašenja u relejnim automatskim sustavima. Svaki strujni krug prikazan je kao pojedinačan odnosno odvojen strujni dio. Strujni dio mora sadržavati barem jedan upravljački uređaj (npr. žarulju, relej ili sl.). Na slikama 4.6. i 4.7. može se vidjeti serijski i paralelni spoj u LAD.



Sl. 4.6. Serijski spoj LAD



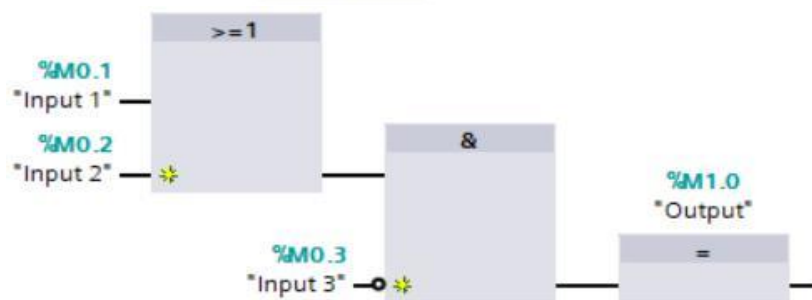
Sl. 4.7. Paralelni spoj u LAD

Na prikazanim slikama može se vidjeti dvije vrste *Inputa*. Ulazni *Inputi* 1 i 2 predstavljaju otvorene sklopke kada na njih ne djeluje signal (eng. *Normally Open Switch*). *Input* 3 predstavlja zatvorenu sklopku kada na nju ne djeluje signal (eng. *Normally Closed Switch*). Ukoliko želimo upaliti *Output* potrebno je dovesti struju do njega, pa je tako u slučaju prikazanom na Slici 4.6. *Inpute* 1 i 2 potrebno je aktivirati, *Input* 3 deaktivirati. U slučaju paralelnog spoja na Slici 4.7. dovoljno je aktivirati samo jedan od *Inputa* 1 ili 2, a deaktivirati *Input* 3.

Električna shema prikazuje stanje kontakta (otvoreno ili zatvoreno), a kod kontaktnog dijagrama ispituje se logičko stanje signala (logička 0 ili logička 1). Logički put unutar LAD koda mora se sastojati od najmanje jedne izlazne naredbe jednoga ili više uvjeta koji moraju biti zadovoljeni da bi se ta naredba ispunila. Oznake Q i I predstavljaju realne fizičke adrese za upravljanje stvarnim procesom. Otvoreni i zatvoreni kontakti zapravo predstavljaju osnovne instrukcije za logičke operacije.

Kompleksniji programi nastaju na temelju kombinacije više logičkih kontakata te tako nastaju logičke funkcije AND, OR, NOT, XOR.

FBD dijagram funkcijskih blokova temelji se na spajanju elektroničkih logičkih sklopova. Kod se zapisuje blokovima funkcija kojima se dodjeljuju ulazne i izlazne varijable. Prednost rada u FBD je veliki izbor blokova te objašnjenje za svaki od njih. FBD je najčešće korišteni programski jezik za korisnike iz razloga što je jednostavan, logičan i pregledan, u slučaju da program ne funkcionira vrlo lako se može pronaći pogreška, slike 4.8. i 4.9. prikazuju primjere serijskog i paralelnog spoja u FBD jezicom.



Sl. 4.9. Paralelni spoj u FBD.

STL (eng. Statement List – listni program) je jezik u kojem se program piše red po red pomoću određenih naredbi. Nedostatak STL-a je nepreglednost i problem što se tek malo napredniji korisnici mogu snaći u programu koji je napisan STL-om. Na slici 4.10. može se vidjeti primjer koda pisanog STL jezikom.

Linija	Kod
A "Tag_Input_1"	//Provjerava je li stanje signala „1“ i ima li AND trenutni RLO
A "Tag_Input_2"	//Provjerava je li stanje signala „1“ i ima li AND trenutni RLO
S „Tag_Output“	//Postavlja se u „1“ ako je RLO 1.“

Sl. 4.10. Primjer STL koda

SCL odnosno strukturirani upravljački jezik je programski jezik koji je strukturom blizak programskom jeziku C zbog sličnih izraza. U slučaju velikog broja izraza SCL zbog svoje forme omogućava preglednosti i jednostavnost u pisanju koda. Na sljedećim slikama 4.11. i 4.12. mogu vidjeti primjeri serijskog i paralelnog SCL koda.

Linija Kod

```
1:        IF "db".i1 = TRUE AND
2:        db".i2 = TRUE AND
3:        db".i3 = TRUE AND
4:        THEN
5:        db".o1 := TRUE;
6:        ELSE
7:        db".o1 :=FALSE
8:        END IF;
```

Sl. 4.11. Serijski primjer SCL koda

Linija Kod

```
1:        IF "db".i1 = TRUE or "db".i2) = TRUE AND
2:        "db".i3 = FALSE
3:        THEN
4        "db".i1 = TRUE
5        ELSE
6        "db".i1 = FALSE
7        END IF;
```

Sl. 4.12. Paralelni primjer SCL- koda.

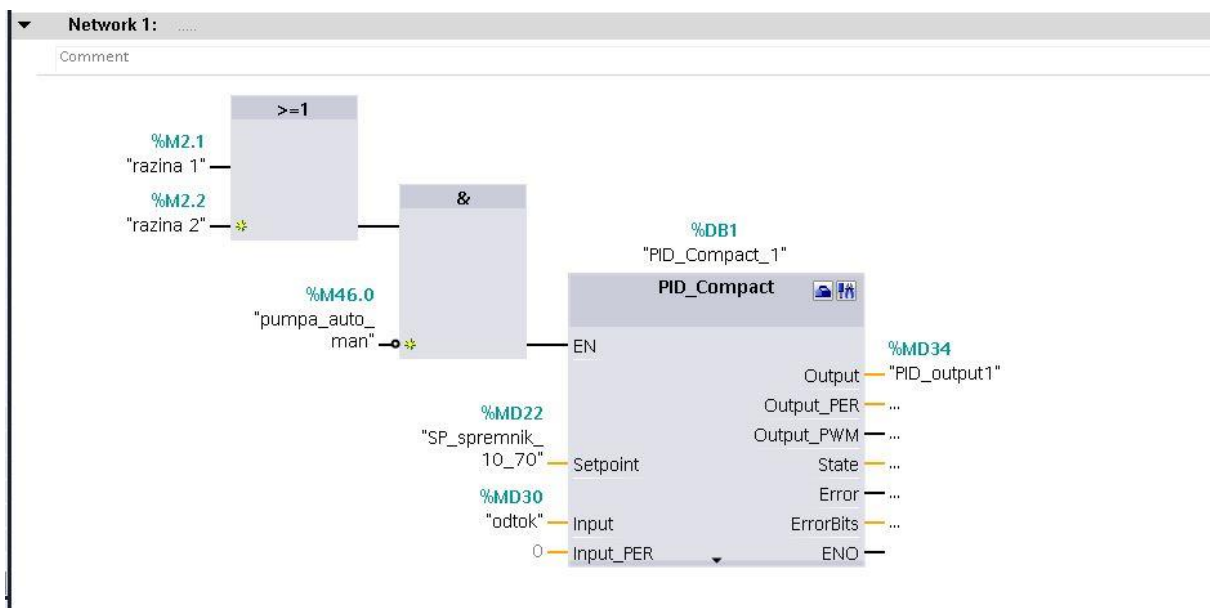
4.2.1. PID Compact

Za korištenje PID regulacije unutar TIA Portala moguće je koristiti ponuđeni programski blok unutar softvera pod nazivom *PID Compact*. Za otvaranje istog potrebno je unutar prozora *Technology* kliknuti na *PID Control* te odabrati *PID_Compact* (vidi sliku 4.13.).

Technology	
Name	Description
▶ Counting	
▼ PID Control	
▼ Compact PID	
▶ PID_Compact	Universal PID controller with integrated tuning

Sl. 4.13. PID Compact

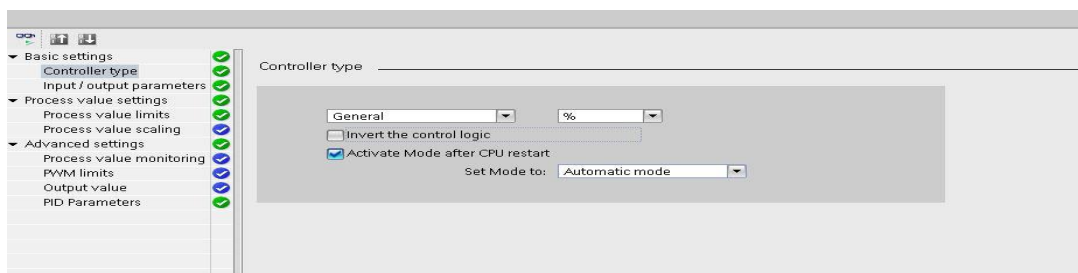
Klikom na *PID Compact* dobijemo prozor u kojemu se nalazi DB (*Data Block*), unutar tog prozora možemo upravljati s željenim varijablama. Varijable kojima možemo upravljati su zadana vrijednost (*Setpoint*), ulazna vrijednost (*Input*) te izlazna vrijednost (*Output*). Isto je prikazano na slici 4.14., slika prikazuje slučaj *PID compact* regulacije procesa odvođenja tekućine iz spremnika pomoću crpke.



Sl. 4.14. PID_Compact_1 unutar TIA portala

Vrijednost ulaza EN (*Enable*) potrebno je postaviti u stanje 1 jer se jedino u tom slučaju blok izvršava zadanu upravljačku logiku.

Nakon toga klikom na plavu ikonicu u gornjem desnom dijelu pod nazivom *Configuration* moramo postaviti *PID Compact* u odgovarajući način rada. Klikom na plavu ikonicu dobije se pristup postavkama veličina i postavkama za izmjenu P,I,D parametara regulatora (vidi sliku 4.15.).



Sl. 4.15. PID Compact configuration

Samopodešavanje *PID compact* regulatora odvija se na sljedeći način. Odabrani programski blok *PID compact* unutar TIA portala potrebno je povezati s pravilno napisanim programom te nakon toga ispravno podesiti regulator. Pomoću opcija *Pretuning* i *Fine tuning* unutar izbornika *Tuning mode* je moguće samopodešavati PID regulator.

Metoda samopodešavanja *Pretuning* zasniva se na Ziegler-Nichols II metodi prijelazne funkcije (vidi detalje u potpoglavlju 2.4.).

Za ispravno podešavanje regulatora potrebno je držati se sljedećih uvjeta:

1. *PID compact* blok mora se pozivati u organizacijskom bloku *Cyclic Interrupt*
2. Ulaz *ManualEnable* mora biti u stanju *False*
3. Ulaz *Reset* također mora biti u stanju *False*
4. *PID compact* blok mora biti u jednom od ponuđenih stanja, *Manual* ili *Automatic mode*
5. Zadana i mjerena veličina moraju biti unutar zadanih granica podešenih u samom bloku
6. Razlika između zadane i mjerene veličine mora biti veća od 50% zadane veličine.

Metoda samopodešavanja *Fine tuning* zasniva se na relejnom postupku podešavanja PID regulatora. Parametri se definiraju na temelju amplitude i frekvencije oscilacija.

Za ispravno podešavanje postavljeni su uvjeti:

1. *PID compact* blok mora se pozivati u organizacijskom bloku *Cyclic Interrupt*
2. Ulaz *ManualEnable* mora biti u stanju *False*
3. Ulaz *Reset* također mora biti u stanju *False*
4. Zadana i mjerena veličina moraju biti unutar granica postavljenih u bloku

5. Regulacijski krug mora biti stabilan u radnoj točki koja se postiže kada su zadana i mjerena veličina jednake
6. Nema očekivanih šumova
7. *PID compact* blok mora biti u jednom od ponuđenih stanja, *Manual ili Automatic mode*

4.2.2. HMI

Gotovo svaki proces u industriji mora nadzirati i kontrolirati čovjek, kako bi on to mogao raditi na što bolji i učinkovitiji način potrebno mu je kontrolu i upravljanje učiniti što jednostavnijim. Isprva su se za kontrolu koristili sinoptički paneli koji su prikazivali shemu procesa, instrumente te pružali zvučne i svjetlosne signale. Takvi paneli nisu u potpunosti zadovoljili potrebe nadzora, radi toga nastala je potreba za daljnjim razvojem standardiziranih sučelja između operatora (čovjeka) i procesa (stroja), odnosno HMI aplikacijama. Zadaci koje obavlja HMI aplikacija su:

- Vizualizacija samog procesa: proces je vidljiv na monitoru, osvježava se i temelji na stvarnim vrijednostima,
- Operator kontrolira proces: čovjek mora imati potpunu kontrolu nad procesom,
- Prikaz upozorenja i alarma: dojava kvara, prekida rada i sl.,
- Arhiviranje procesnih vrijednosti: bilježe se sve vrijednosti uključujući alarme radi bolje kontrole kvalitete ,
- Obrada procesnih vrijednosti: moguća statistička obrada.

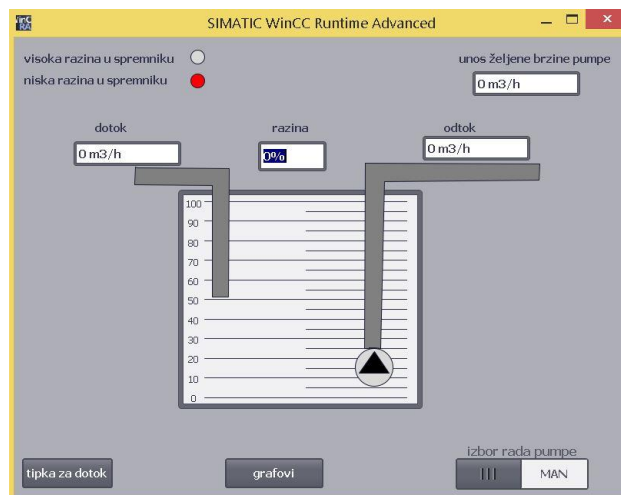
Na slici 4.16. može se vidjeti primjer HMI sučelja koje je razvijeno u okviru ovog rada. Pod HMI aplikacijom podrazumijeva se vizualizacija na više načina, od običnih tekstualnih panela na integriranim PLC-ovima, do velikih, više-računalnih upravljačkih soba. Odabir uređaja i aplikacije ovisi o primjeni.

Danas osim osnovne računalne opreme za nadzor procesa (računalo s HD monitorom, tipkovnicom i mišem) koriste se i touch screen paneli, svjetlosna pera, HDD-ovi, vanjska memorija, pisači, čitači magnetskih kartica, čitači otiska prsta, operatorski paneli i slično.

Izradom SCADE riješen je problem složenosti nadzora, osim što omogućava nadzor procesa sustav SCADE prikuplja sve vrste podataka, obrađuje podatke te stvara baze za njihovu pohranu. Sustav vrši obradu svi događaja, upozorenja, alarma te o njima izvještava korisnika,

korisnik u bilo kojem trenutku ima uvid u grafički prikaz stanja sustava. Unutar velikih industrijskih pogona javlja se potreba za kontrolom određenih dijelova pogona, pošto je postrojenje veliko dijelovi nalaze se daleko jedan od drugog. Potreban nam je nadzor s SCADOM i HMI, razlika je u njihovom opsegu. SCADA se koristi za reguliranje crpki, ventilatora i ostalih strojeva, koji se reguliraju pomoću PLC-ova. PLC kontrolira rad stroja, informacije se šalju u kontrolnu sobu (vidi sliku 4.17.) gdje im se „daje“ smisao i naredba o potrebnoj radnji. To mjesto predstavlja HMI, koji prikazuje čitav sustav s kontrolama, sensorima, mjeracima, oznakama i alarmima smještenim na strojevima. Dakle, SCADA je potpuni kontrolni sustav, dok ljudski faktor djeluje putem HMI-a.

PLCSIM je dio softvera TIA Portal koji se koristi za simulaciju napisanog koda, simulacija se izvršava na način da se program preuzme na virtualni uređaj. Alat omogućava testiranje i isprobavanje napisanog programa, uočavanje eventualnih grešaka prije puštanja u pogon



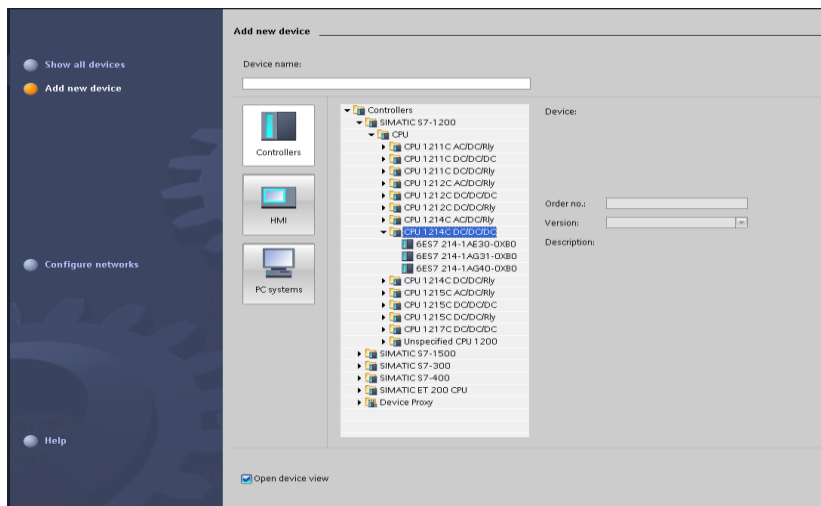
Sl. 4.16. Primjer HMI sučelja



Sl. 4.17. Primjer kontrolne sobe.

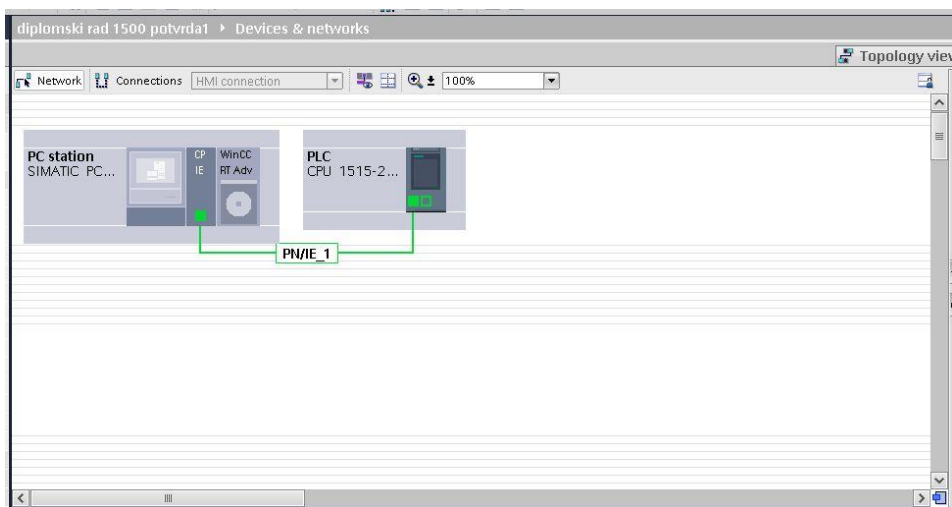
5. SIMULACIJA I VIZUALIZACIJA PROCESA USKLADIŠTENJA U OKVIRU TIA PORTALA

Nakon pokretanja TIA Portala potrebno je dodati željene uređaje klikom na „Devices and Networks“ pa potom „Add new devices“ gdje možemo odabrati koji uređaj želimo dodati.



Sl.5.1. Odabir željenog PLC-a.

Prilikom izrade ovog rad izabran je PLC1200 1214C DC/DC/DC 6ES7-214-1AG40-0XB0. Slika 5.1. prikazuje prozor unutar kojeg se odabire željeni PLC. Nakon odabira željenog PLC-a otvara se „Device view“ gdje se vidi odabrani PLC. U toj alatnoj traci mogu se konfigurirati i povezivati uređaji s pojedinim modulima (vidi sliku 5.2.).



Sl. 5.2. Devices and networks unutar TIA portala

U ovom radu dodan je komunikacijski model odnosno mrežna kartica *PROFINET/Ethernet* > *IEgeneral te Simatic HMI application-WinCC RT Advanced*. WinCC je softver koji se koristi za vizualizaciju i izvještaj HMI aplikacije.

Dodavanjem bilo kojeg PLC-a pojavljuje se glavni organizacijski blok „Main“ koji sadrži program korisnika. Svaki korisnički program mora imati najmanje jedan organizacijski blok unutar kojeg se mogu nalaziti različite funkcije ovisno o potrebnim programima. Pod kraticom *Program blocks* nalazi se *Main* unutar kojeg se piše kod. Najvažni programski blok je blok OB1 te se izvršava se ciklički. Program organizacijskog bloka podijeljen je u mreže, programi se mogu pisati u različitim jezicima kao što su LAD, FBD, STL i STCL. Svaki od njih detaljno je objašnjen u potpoglavlju 4.2.

Za dodavanje PID compacta dodan je blok *Cycil interrupt* koji se koristi u slučaju kada želimo da se program izvršava u zadanim vremenskim razmacima, neovisno o cikličkom izvršavanju.

Varijabla koju koristimo za sve vrijednosti je tag, tagove dijelimo na dvije vrste, a to su lokalni i PLC tagovi. Lokalni tagovi se primjenjuju samo u bloku u kojem su definirani, a PLC tagovi se primjenjuju na cijelom PLC-u. Slika 5.3. prikazuje tablicu s tagovima koji se koriste u TIA portalu za proces odvođenja tekućine iz spremnika pomoću crpke.

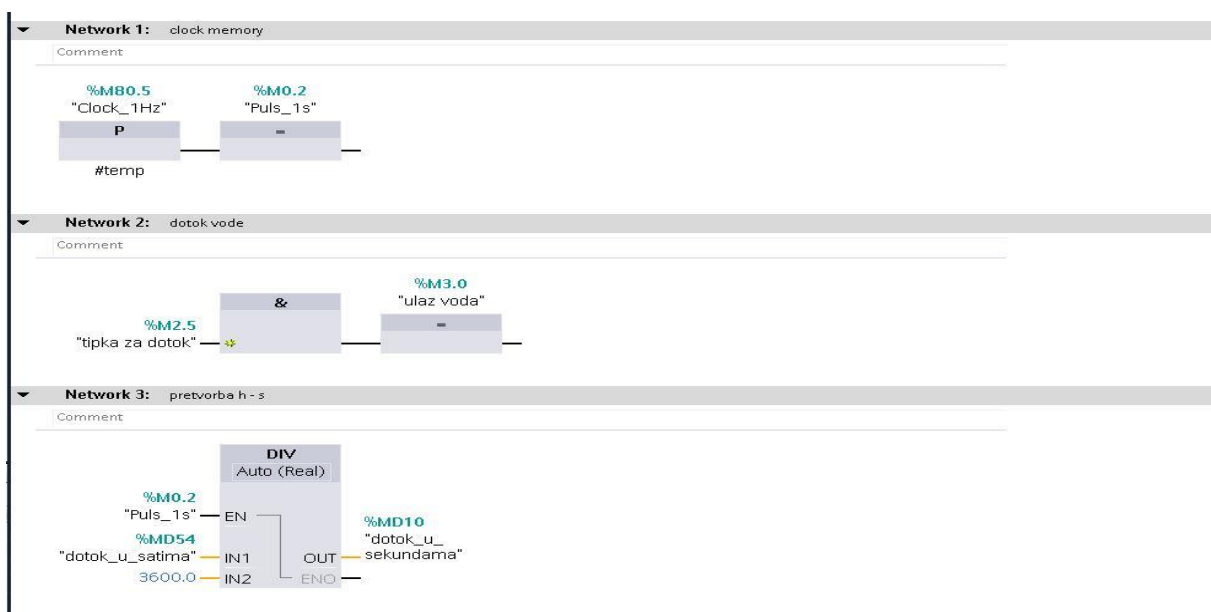
Name	Data type	Address	Retain	Visibl...	Acces...	Comment	
4	AlwaysTRUE	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	AlwaysFALSE	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	razina 0	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	razina 1	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	razina 2	Bool	%M2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	razina 3	Bool	%M2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	razina 4	Bool	%M2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	tipka za dotok	Bool	%M2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	ulaz voda	Bool	%M3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	brojac	Real	%MD6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	pumpa start	Bool	%M3.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	najmanja razina	Bool	%M3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	najveća razina	Bool	%M3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	dotok_u_sekundama	Real	%MD10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Clock_Byte	Byte	%MB80	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	Clock_10Hz	Bool	%MB0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	Clock_5Hz	Bool	%MB0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	Clock_2.5Hz	Bool	%MB0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Clock_2Hz	Bool	%MB0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Clock_1.25Hz	Bool	%MB0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	Clock_1Hz	Bool	%MB0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	Clock_0.625Hz	Bool	%MB0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	Clock_0.5Hz	Bool	%MB0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	razina_u_spremniku	Real	%MD14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	SP_spremnik_10_70	Real	%MD22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	SP_spremnik_70	Real	%MD26	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	odtok	Real	%MD30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	PID_output1	Real	%MD34	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	PID_output2	Real	%MD38	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	brzina pumpe	Real	%MD42	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	pumpa_automat	Bool	%M46.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	hmi_brzina	Real	%MD50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	dotok_u_satima	Real	%MD54	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37	protok_senzor	Real	%MD58	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38	podjela	Real	%MD62	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39	izlaz_pumpe_m3h	Real	%MD66	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	izlaz_iz_spremnika_m3	Real	%MD70	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41	LED_visoka_razina	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42	LED_niska_razina	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43	Puls_1s	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Sl. 5.3. Tablica s tagovima unutar TIA portala

5.1. Implementacija procesa uskladištenja tekućine i regulacije u FBD jeziku

Na slici 5.4. se nalaze tri mreže napisanog koda:

- U mreži 1 omogućava se sistemska memorija odnosno korištenje stvarnog sata PLC-a radi simulacije programa.
- U mreži 2 AND blokom je definirano da ukoliko je aktivan tag „tipka za dotok“ voda će početi ulaziti u spremnik.
- Unutar mreže 3 nalazi se pretvorba dotoka vode iz m^3/h u m^3/s pomoću bloka DIV, pretvorbu je potrebno napraviti radi kasnije linearizacije procesa

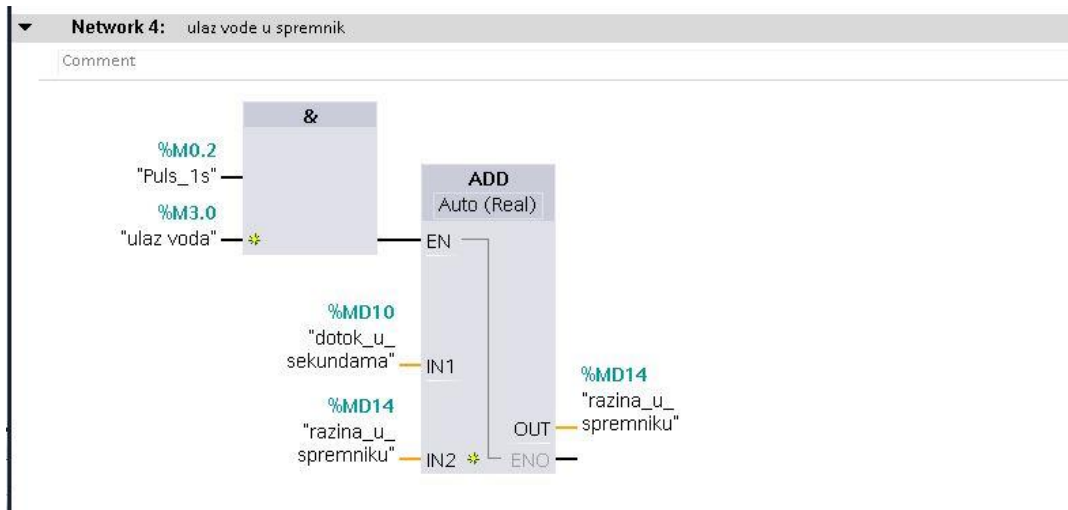


Sl. 5.4. Prikaz network 1,2 i 3 unutar TIA portala

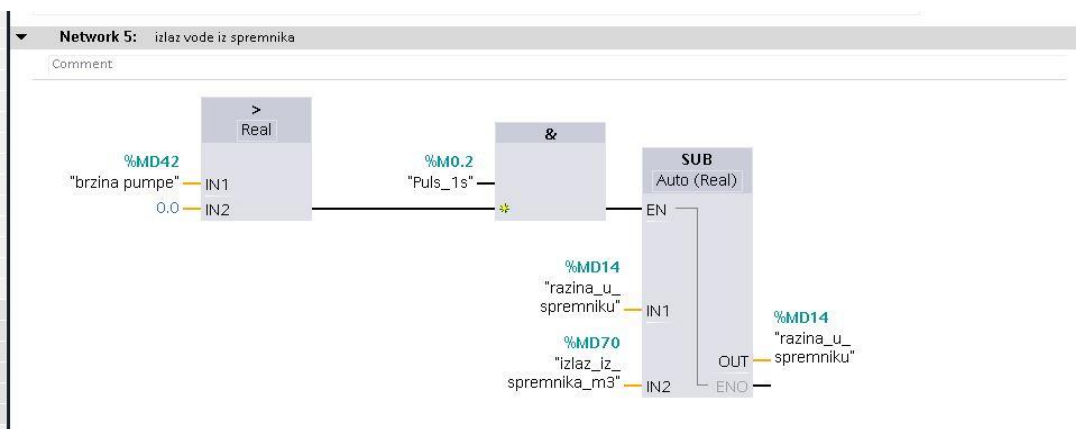
Na slici 5.5 prikazana je mreža 4:

- Logika mreže 4 nam govori da ukoliko je aktivan bit „ulaz voda“ pomoću matematičkog bloka ADD(zbrajanje) dolazi do punjenja spremnika na način da se vrijednosti tagova „dotok u sekundama“ i „razina u spremniku“ međusobno zbrajaju svake sekunde te se zbrojena vrijednost sprema u varijablu „razina u spremniku“.

- Na sličan način dolazi do izlaza vode iz spremnika u mreži 5 na slici 5.5., pomoću matematičkog bloka SUB (oduzimanje) od vrijednosti „razina u spremniku“ oduzima se vrijednost „izlaz iz spremnika“



Sl. 5.5. Prikaz network 4 unutar TIA portala

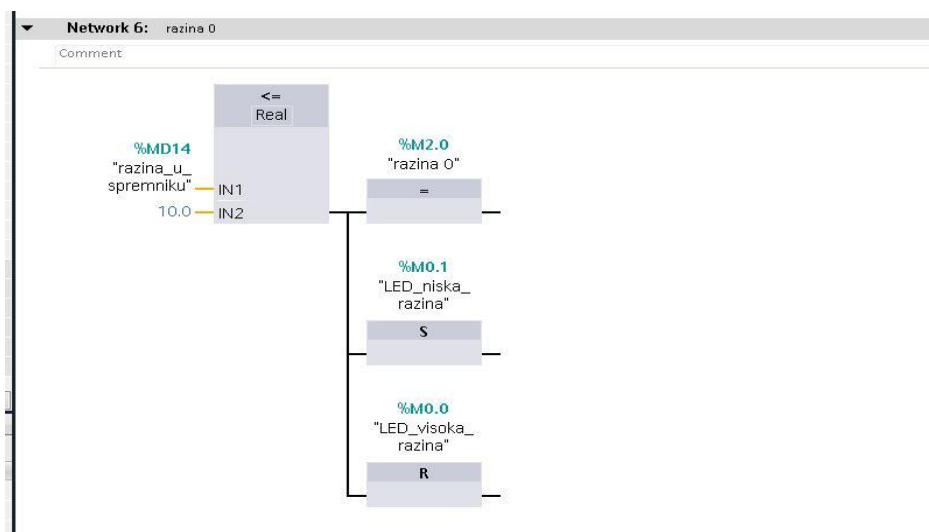


Sl. 5.6. Prikaz network 5 unutar TIA portala.

Na slici 5.7 nalazi se mreža 6:

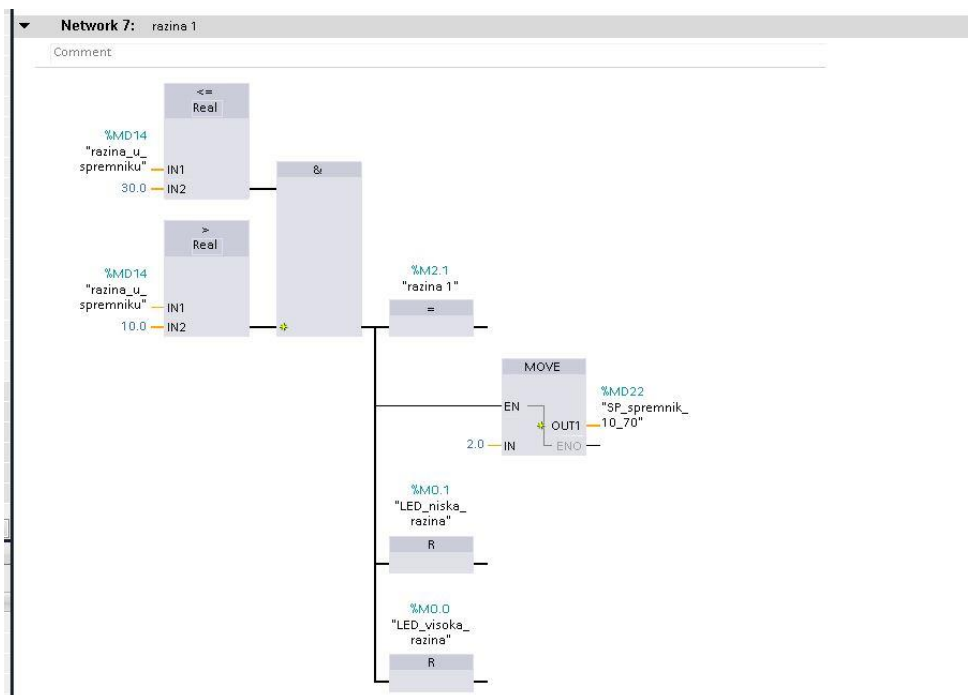
- definirana je razina 0 (vidi sliku 5.7.), za istu vrijedi da je razina vode u spremniku između 0 i 10 posto spremnika. Razina je definirana pomoću komparatora odnosno matematičke operacije *LESS OR EQUAL*. Bit „LED_niska_razina“ je uključen

pomoću bloka SET, a koristimo ga za aktivaciju alarma „*niska razina*“. Tag „*LED_visoka_razina*“ potrebno je resetirati.



Sl. 5.7. Prikaz network 6 unutar TIA portala

- U mreži 7 na slici 5.8. definirana je razina 1, pa tako razina vode za istu mora biti između 10 i 30. Kao u prethodnom slučaju razine smo definirali pomoću matematičkih komparatora *LESS OR EQUAL* i *GREATER OR LESS*. Pomoću

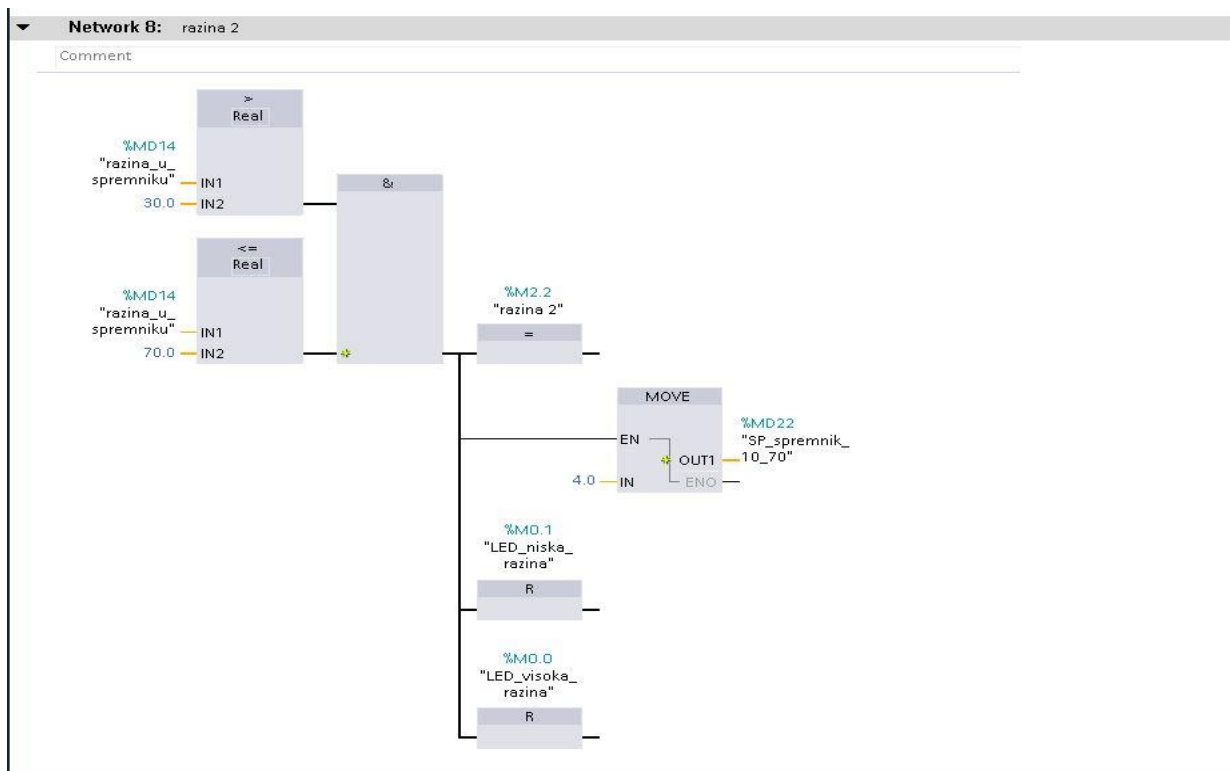


Sl. 5.8. Prikaz network 7 unutar TIA portala.

bloka *MOVE* u varijablu „*SP_spremnik_10_70*“ sprema se vrijednost 2 kao setpoint za *PID_compact_1*. Varijable s alarmima su resetirane.

Razinu 2 definirana je prema slici 5.9. unutar mreže 8:

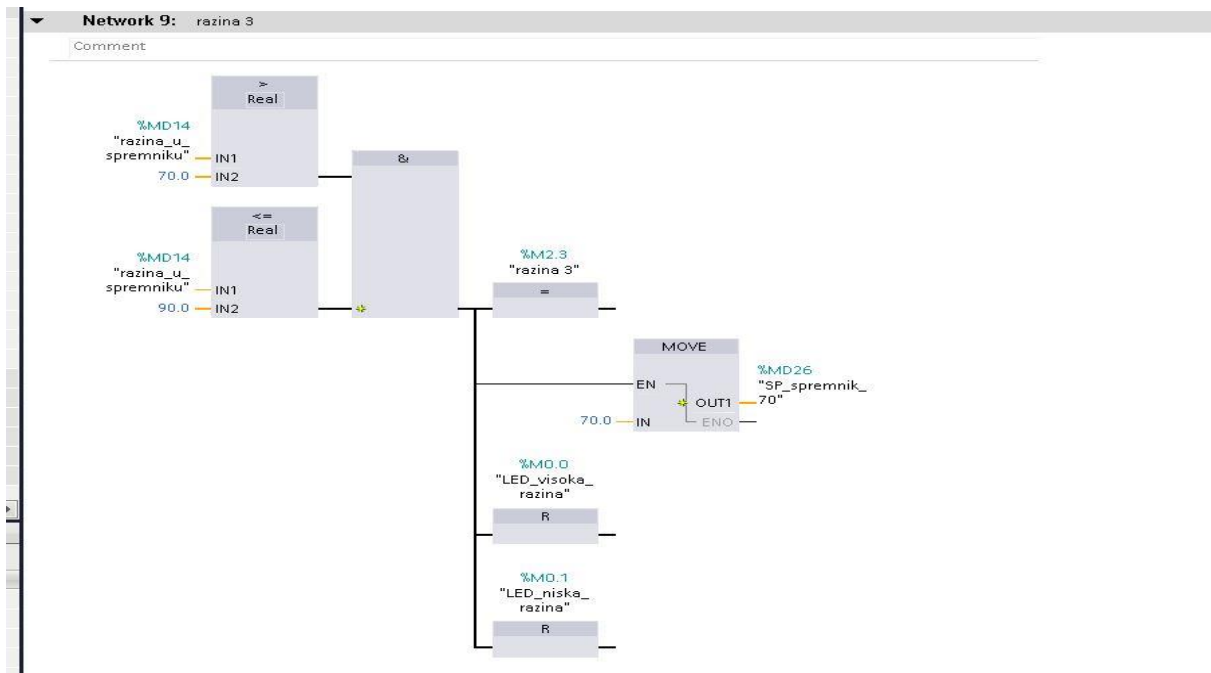
- Pomoću komparatora i matematičke operacije definiran je raspon u razini vode u spremniku, pa je tako za razinu 2 voda u spremniku od 30 do 70 posto spremnika. Prema sličnoj logici kao u prijašnjem slučaju vrijednost 4 pomoću bloka *MOVE* prenesena je na tag „*SP_spremnik_10_70*“ koji unutar *cycle interrupt* označava setpoint *PID_compacta_2*.



Sl. 5.9. Prikaz network 8 unutar TIA portala.

Razina 3 koja je definirana na slici 5.9.:

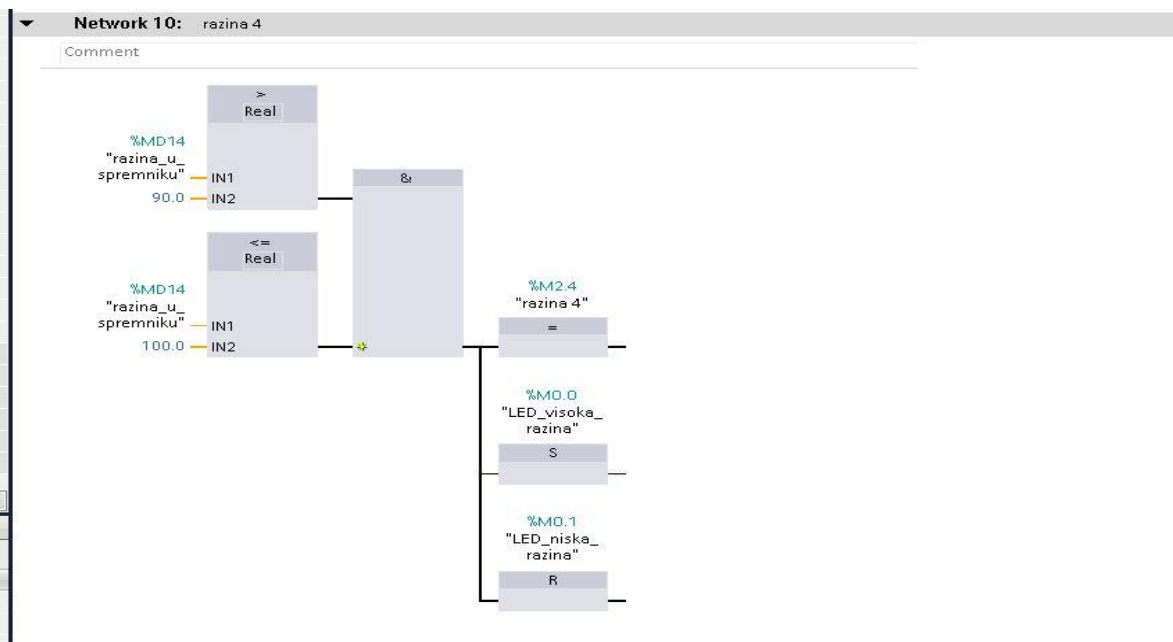
- mrežom 9 pomoću bloka *MOVE* vrijednost 70 je predana varijabli „*SP_spremnik_70*“ koja unutar *cycle interrupt* predstavlja setpoint *PID_compact_2*.



Sl. 5.10. Prikaz network 9 unutar TIA portala.

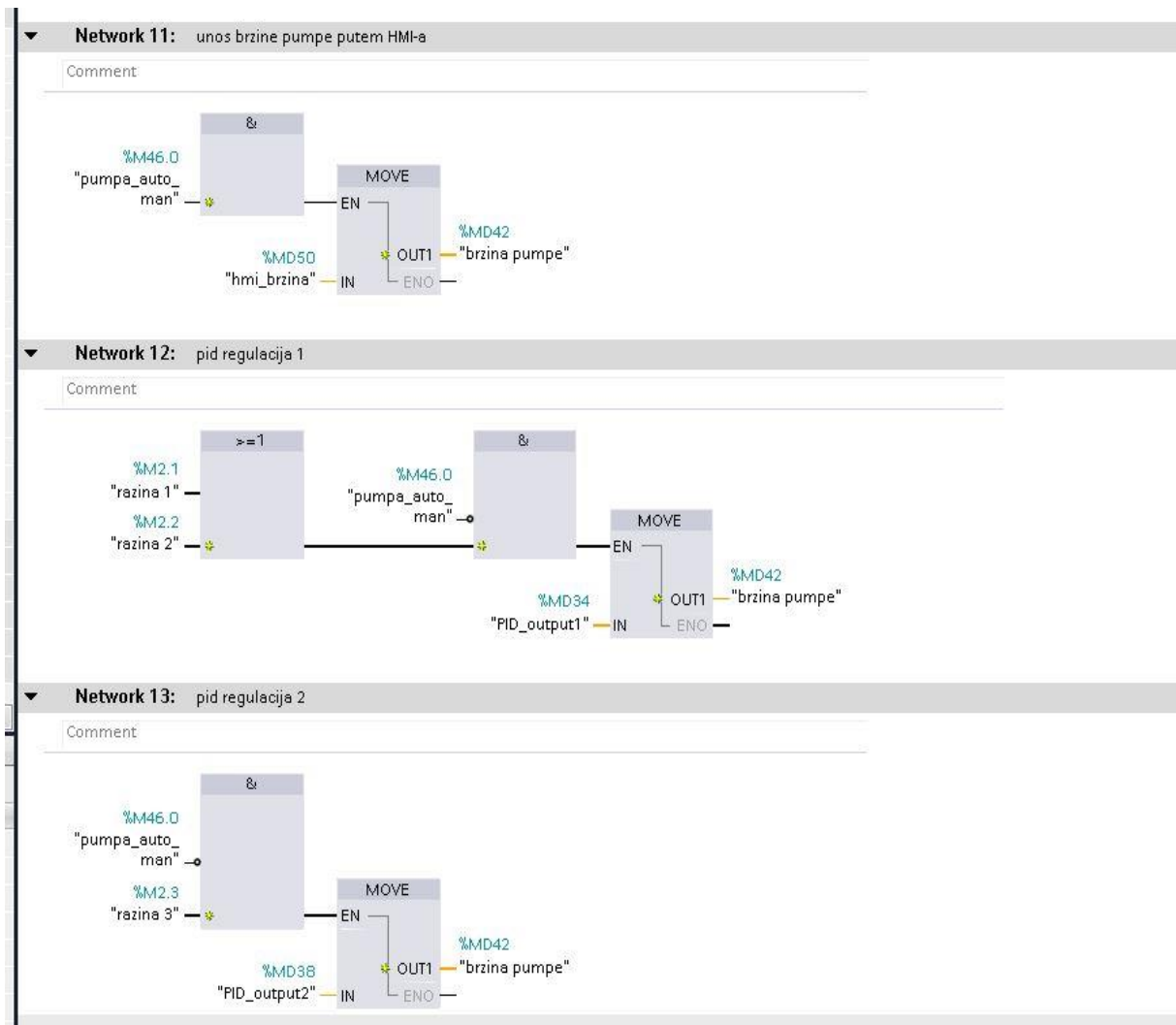
Mreža 10 prikazana je na slici 5.11.:

- Logikom kao i u prijašnjim slučajevima definirana je posljednja razina 4, kada je razina vode između 90 i 100. Samim time i tag alarma „LED_visoka_razina“ je potrebno setirati.



Sl. 5.11. Prikaz network 10 unutar TIA portala.

Program će signalizirati nisku odnosno visoku razinu pomoću crvene lampice koja će se paliti na sučelju.



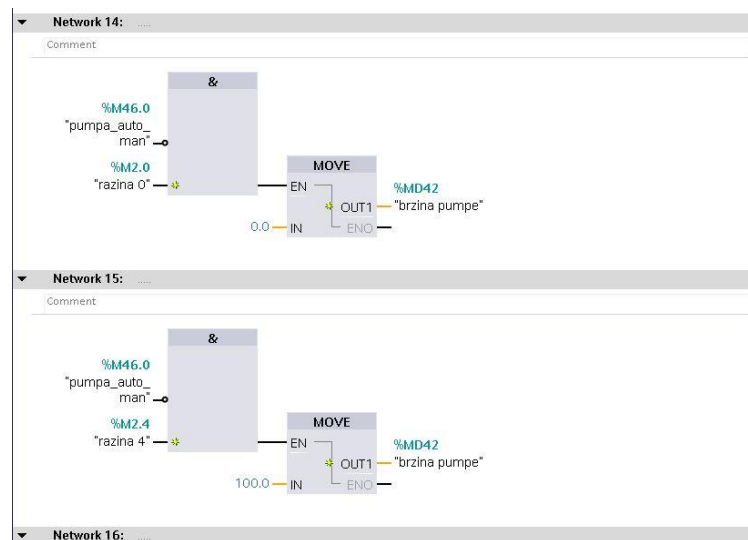
SI 5.12. Prikaz network 11,12 i 13 unutar TIA portala.

Na Slici 5.12. nalaze se mreža 11, mreža 12 i mreža 13:

- U mreži 11 pomoću taga „*pumpa_auto_man*“ na HMI sučelju može se odabrati automatski ili ručni način rada. Pomoću taga „*hmi brzina*“ osoba na HMI sučelju može upisati željnu brzinu crpke.
- U mreži 12 definirano je ukoliko je aktivana razina 1 ili razina 2 te tag „*pumpa_auto_man*“ nije aktivan brzina crpke biti će određena PID compactom odnosno PID regulacijom.
- U mreži 13 nalazi se slična logika ukoliko je aktivna razina 3 i tag „*pumpa_auto_man*“ nije aktivan brzina crpke određena je PID regulacijom.

Slika 5.1. prikazuje mreže 14 i 15:

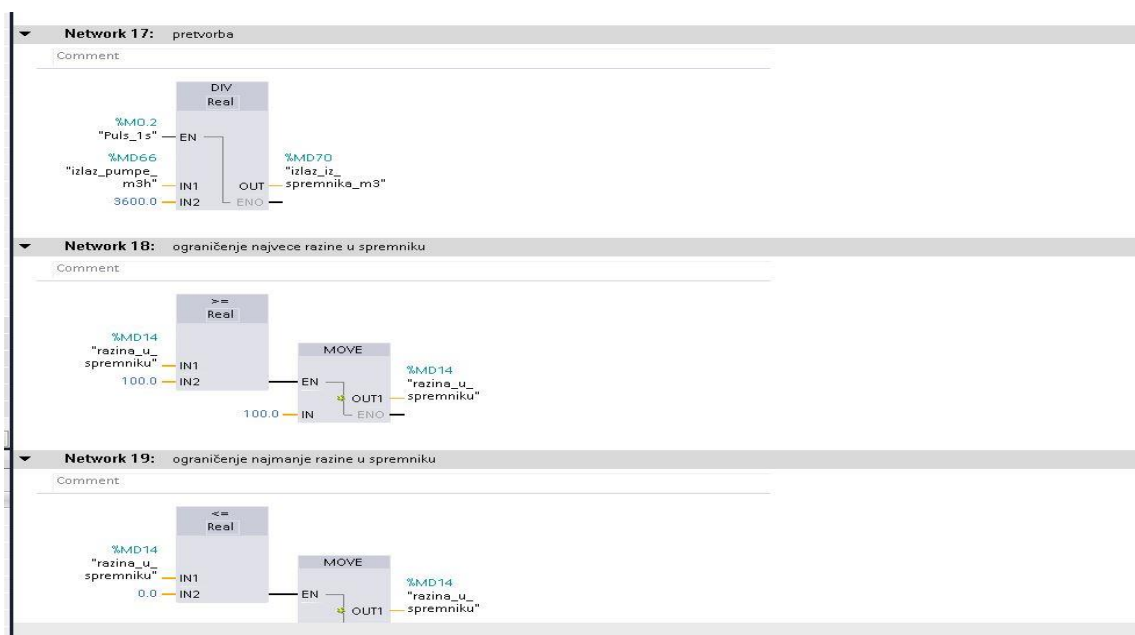
- Unutar mreže 15 pomoću negacije se definira ručni rad crpke. Mreža 15 definira rad crpke u trenutku kada je aktivna razina 4, tada crpka radi maksimalnom brzinom.



Sl. 5.13. Prikaz network 14 i 15 unutar TIA portala

Na slici 5.14. nalaze se mreže 17,18,19:

- Mrežom 17 ponovo vršimo pretvorbu iz m^3 / h u m^3 / s . Dok se u mrežama 18 i 19 pomoću bloka *MOVE* ograničava maksimalna i minimalna razina u spremniku kako bi se osigurao ispravan rad programa (vidi sliku 5.14.).



Sl. 5.14.. Prikaz network 17,18,19 unutar TIA portala.

Slika 5.15. prikazuje mreže 20 i 21:

- U obje mreže vrši se pretvorba mjernih jedinica varijabli odnosno kako bi dobiveni rezultati bili u odgovarajućim mjernim jedinicama.



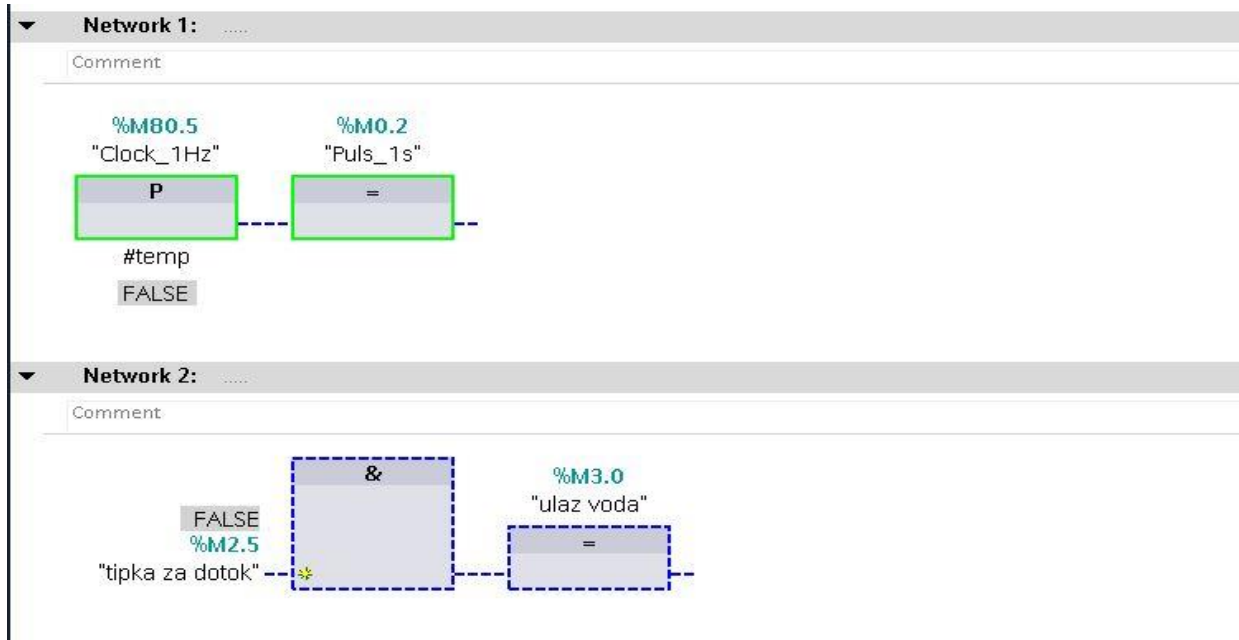
Sl. 5.15.. Prikaz network 19 20 i 21 unutar TIA portala.

5.2. Pokretanje programa u PLC-u i vizualizacija pomoću HMI

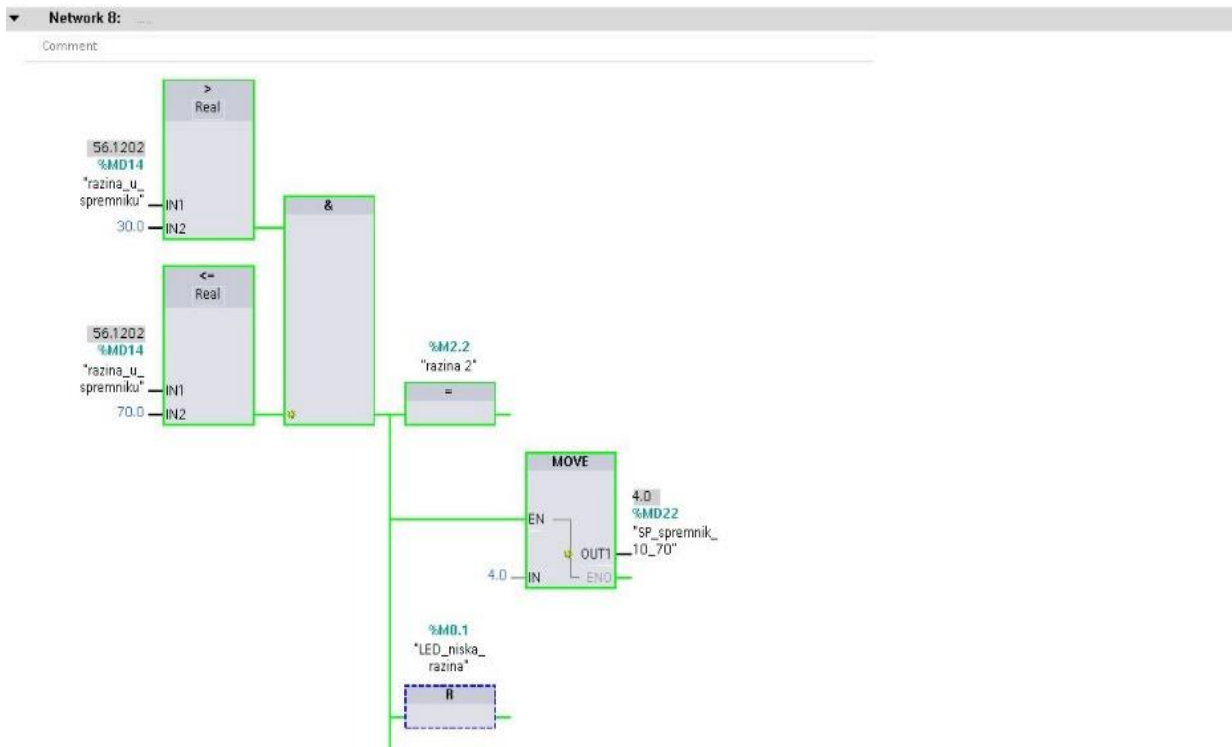
Na slici 5.16. može se vidjeti početak simulacije programa, u trenutku dok *tipka za dotok* na HMI sučelju nije stisnuta, voda ne ulazi u spremnik. Pritiskom na tipku započinje proces simulacije, spremnik se počinje puniti vodom. Vidi se da tag *tipka za dotok* prelazi iz stanja *false* u stanje *true* pritiskom tipke na HMI sučelju. HMI sučelje prikazano je na slici 5.19..

Slika 5.17. prikazuju dio programa u kojem dolazi do definiranja razine vode u spremniku. Unutar koda nalazi se alarmi koji signaliziraju visoku odnosno nisku razinu vode u spremniku.

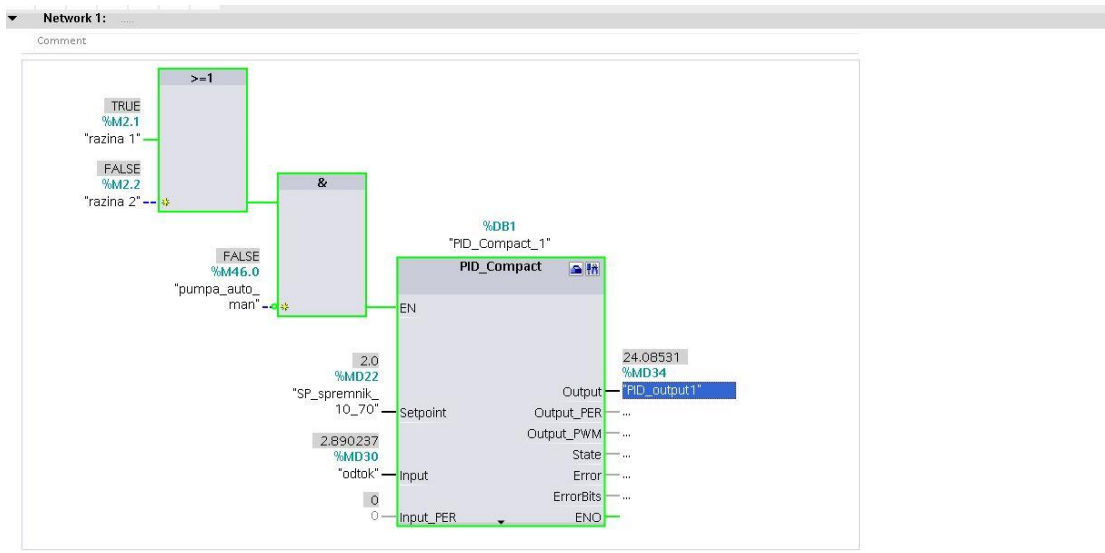
Dolaskom vode do razine 1 započinje PID regulacija prema kojoj ukoliko je razina vode između 10% i 30% spremnika crpka ispumpava vodu s ciljem održavanja protoka 2m/s. Iz tog razloga kao setpoint postavljena je vrijednost 2.



Sl. 5.16. Prikaz početka simulacije u TIA portalu



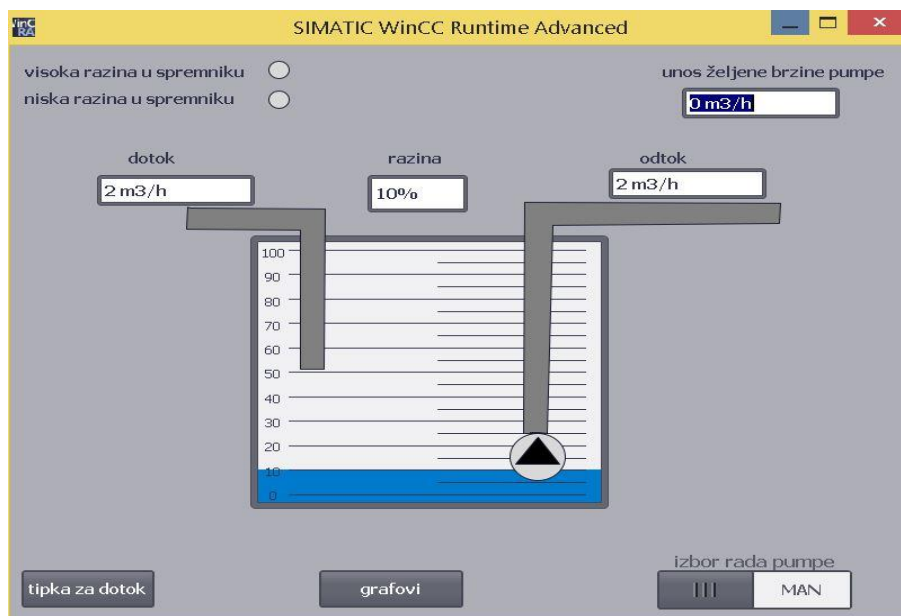
Sl. 5.17. Prikaz network 8 tijekom simulacije unutar TIA portala



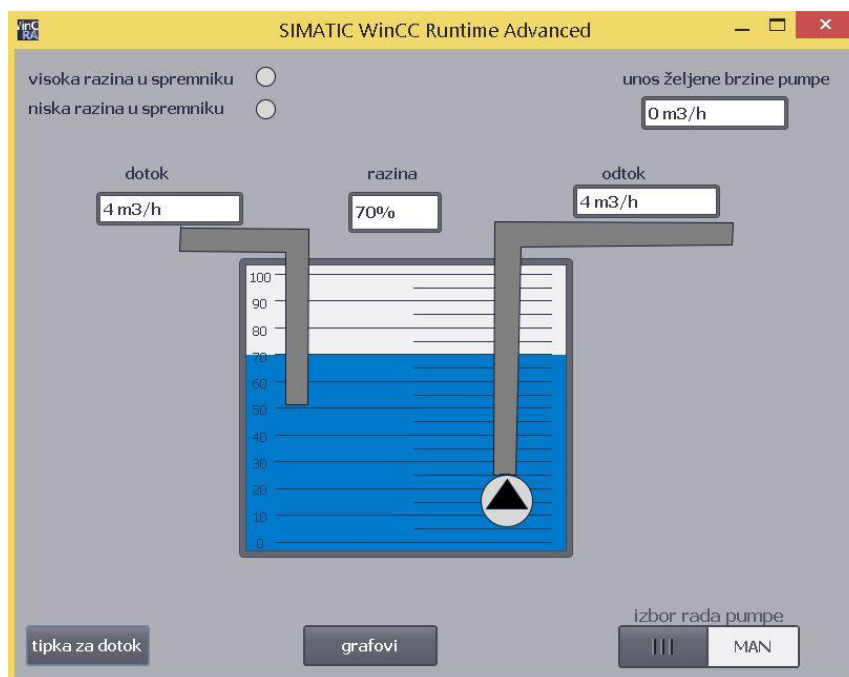
Sl. 5.18. Prikaz PID_compact_1 tijekom simulacije unutar TIA portala.

Ispred stavke EN (*Enable*) nalaze se uvjeti koji moraju biti zadovoljeni kako se se PID Compact izvršio. Slično se dogodi kad se razine vode nalazi između 30% i 70% posto spremnika, tada je kao setpoint postavljena vrijednost 2. Na slici 5.18. može se vidjeti *PID_Compact_1* u trenutku regulacije.

U slučaju kada je razina vode u spremniku između 70% i 90% započinje regulacija crpke s ciljem održavanja razine 70% pomoću *PID_Compact_2* (vidi sliku 5.20.).

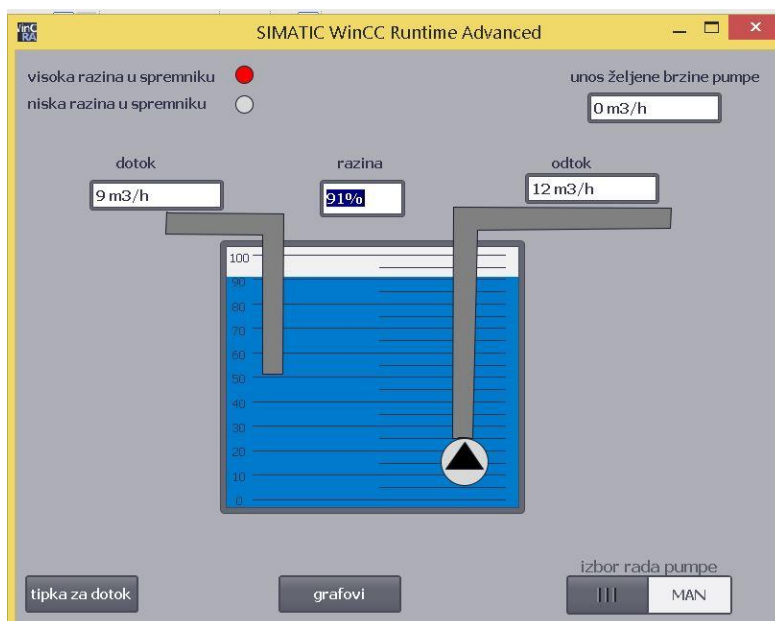


Sl. 5.19. Prikaz HMI sučelja u tijekom simulacije unutar TIA portala.



Sl. 5.20. Prikaz HMI sučelja tijekom simulacije unutar TIA portala

Ukoliko je razina vode popne iznad 90% spremnika aktivirati će se alarm odnosno crvena lampica na HMI sučelju, brzina crpke tada će biti maksimalna. (vidi sliku 5.21.).

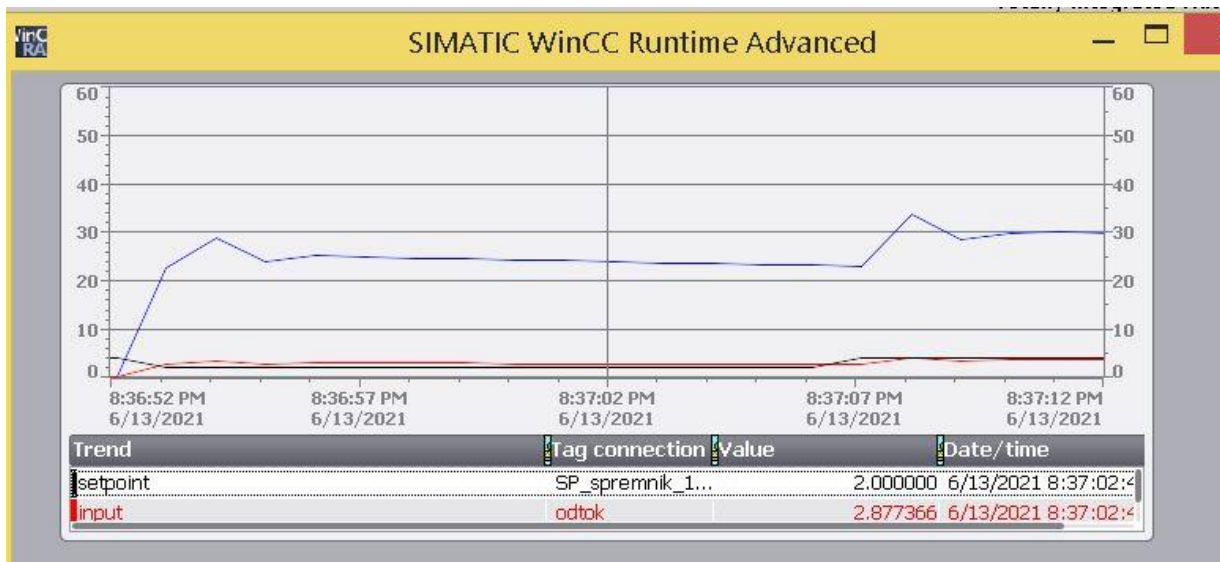


Sl. 5.21. Prikaz HMI sučelja tijekom simulacije unutar TIA portala

Unutar programa korisniku je omogućeno da odabire način rada crpke, automatski rad crpke postavljen je kao zadani. Pritiskom na tipku „MAN“ na HMI sučelju crpka počinje raditi u

ručnom načinu rada, korisnik na HMI sučelju može unijeti željenu brzinu crpke. Unutar koda potrebno je ograničiti unos brzine kako ne bi došlo neispravnog rada programa ili oštećenja crpke.

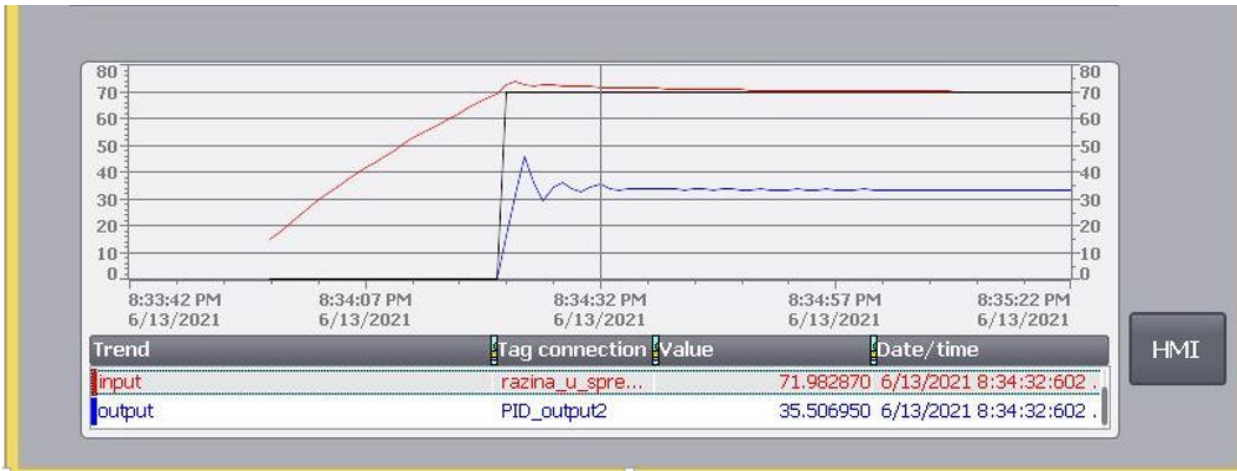
Pritiskom na tipku *grafovi* korisnik ima uvid u grafičke odzive regulacije koje će se obraditi u poglavlju odzivi mjerenja. Grafički prikaz na 5.22. prikazuje odziv regulacije prema *PID_Compact_1*. Iz grafa možemo vidjeti promjenu vrijednosti outputa prilikom regulacije.



Sl. 5.22. Prikaz odziva PID compact_1 unutar TIA portala

Na slici 5.23. može se vidjeti odziv prema *PID_Compact_2*. Na oba grafička prikaza plavom bojom označano je krivulja *outputa*. Crna krivulja prikazuje vrijednost *setpointa*, a crvena vrijednost *inputa*. U oba slučaja jasno se može vidjeti djelovanje PID regulatora odnosno da smo dobili željene odzive.

Kada pogledamo rezultate simulacija jasno se vidi regulaciju prema PID regulatoru u trenucima koji su definirani kodom. Prilikom prve simulacije kada nisu bili definirani PID parametri pojavljivale su se velike i štetne oscilacije. Korištenjem metoda Ziegler Nichols za određivanjem parametara PID regulatora dobiveni su novi parametri koji su ubačeni u postavke PID Compacta. Pri početnoj vrijednosti se pojavljuju blage oscilacije, ali amplituda nije velika. Oscilacije se ubrzo smire i vrijednost *outputa* se ustali. P dio regulatora dobiven je uz pomoć Ziegler - Nichols metode korištene za simulaciju blokovskog dijagrama u Simulinku.



Sl. 5.23. Prikaz odziva PID_compact_2 unutar TIA portala

6. ZAKLJUČAK

Tehnologija danas svakim danom napreduje, napretkom tehnologije povećavaju se potrebe za novim rješenjima u industriji. Automatizacija će se morati razvijati što više te će roboti i strojevi sve više zamjenjivati ljude. Čovjek već danas pomoću HMI sučelja može imati vizualizirani prikaz procesa i upravljati samim procesom.

Ovaj rad obuhvaća kompletan proces modeliranja uskladištenja tekućine u spremniku s crpkom unutar programskog paketa TIA Portal i MATLAB. Prikazan je postupak dodavanja PLC-a i potrebnih uređaja, pisanje koda te pokretanje vizualizacije i simulacije cijelog procesa. HMI odnosno nadzorno- upravljačka jedinica pojednostavljuje i olakšava korisniku rad s procesom. Napravljen je matematički model te blokovski dijagram za proces unutar programskog paketa MATLAB. Na temelju Ziegler – Nichols metode određeni su parametri PID regulatora.

Najveća prednost korištenja TIA Portala je jednostavnost i preglednost prilikom rada što korisniku smanjuje vrijeme potrebno za izradu projekata.

LITERATURA

- [1] Joško Petrić, Automatska regulacija: uvod u analizu i sintezu , Fakultet strojarstva i brodogradnje u Zagrebu 2012, dostupno na:
http://titan.fsb.hr/~jpetric/Udzbenici/Udzbenik_AUTOMATSKA%20REGULACIJA_JPetric.pdf
(10.05.2021)
- [2] Prof.dr.sc. Joško Petrić, Mihael Cipek, dipl.inž, Osnove automatizacije, dostupno na:
https://titan.fsb.hr/~mcipek/base/novosti/oa_pi_I_dio.pdf (10.5.2021.)
- [3] Prof.dr.sc. Nedjeljko Perić, Prof. dr.sc. Ivan Petrović, Doc. dr. sc. Mario Vašek, Procesna automatizacija, FER, 2013, dostupno na:
[https://www.fer.unizg.hr/_download/repository/Skripta_Proaut_veljaca_2013\[1\].pdf](https://www.fer.unizg.hr/_download/repository/Skripta_Proaut_veljaca_2013[1].pdf) (10.5.2021)
- [4] dip.ing Goran Malčić, Programirljivi logički kontroleri, Tehničko veleučilište u Zagrebu,elektrotehnički odjel, dostupno na:
https://kupdf.net/download/plc-skripta-tvz_5b08a8eae2b6f5ff70117aa9_pdf (5.6.2021)
- [5] Dokumentacija iz tvrtke TEO-Belišće, Simatic S7 tečaj, TIA_BASIS_v15.11_electronic (10.5.2021)
- [6] http://www.plcdev.com/how_plcs_work.(5.6.2021)
- [7] Siemens, Simatic S7–1200,S7-1500PID control, 2017.(15.5.2021)
- [8] <https://www.mathworks.com/help/simulink/> (10.8.2021.)
- [9]<https://www.plc-city.com/shop/en/siemens-simatic-s7-1200-cpu-1212c/6es7212-1be31-0xb0.html> (10.8.2021)
- [10] <https://www.mathworks.com/products/matlab.html> (10.8.2021.)

SAŽETAK

Tema diplomskog rada je izrada upravljačkog sustava za regulaciju procesa uskladištenja tekućine u spremniku pomoću PLC-a. Unutar programa MATLAB izvršen je simulacija programa na temelju matematičkog modela, te su prikazani dobiveni odzivi. Unutar softvera TIA Portal odabran je PLC, izvršeno spajanje s uređajima koji omogućuju simulaciju, te je izrađeno HMI sučelje za regulaciju procesom. Napisan je kod procesa u FBD, te je svaka mreža koda detaljno objašnjena. Prikazani su i objašnjeni dobiveni odzivi regulacijskog kruga.

Ključne riječi: TIA Portal, HMI, MATLAB, spremnik s vodom, PID regulator

ABSTRACT

The main purpose of this paper is to develop a control system for water regulation in the tank using a PLC program. Thus, with the help of the MATLAB program, a program simulation based on a mathematical model was performed and the obtained responses are presented in the paper. Furthermore, within the TIA Portal software, the appropriate PLC was selected and the connection with the devices that enable this simulation was performed, users are enabled to manage this process using the HMI interface. Based on this, it was written a FBD process code and all the networks are extensively explained.

Keywords: TIA Portal, HMI, MATLAB, water tank , PID controller

ŽIVOTOPIS

Luka Parmačević rođen je 3.lipnja 1996. u Osijeku. Nakon završene Osnovne škole Ladimirevci u Ladimirevcima upisao je Srednju školu Valpovo u Valpovu smjer opća gimnazija. Srednju školu završava sa vrlo dobrim uspjehom te iste godine polaže državnu maturu. Nakon državne mature 2015. godine upisuje preddiplomski sveučilišni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Informatički je pismen te se izvrsno služi programskim paketom Microsoft Office (Excel, Word, PowerPoint) te programskim paketom AutoCAD. Cilj mu je po završetku fakulteta zaposliti se u struci.

U Osijeku, 15. rujna 2021.

Luka Parmačević

Potpis: