

# Sekvencioner za pokretanje hidrauličke jedinice

---

Strinić, Luka

Undergraduate thesis / Završni rad

2018

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:200:751485>

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

Download date / Datum preuzimanja: **2024-11-20**

Repository / Repozitorij:

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



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

**Stručni studij**

**SEKVENCIONER ZA POKRETANJE HIDRAULIČKE  
JEDINICE**

**Završni rad**

**Luka Strinić**

**Osijek, 2018.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju**

Osijek, 18.09.2018.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju**

<b>Ime i prezime studenta:</b>	Luka Strinić
<b>Studij, smjer:</b>	Preddiplomski stručni studij Elektrotehnika, smjer Automatika
<b>Mat. br. studenta, godina upisa:</b>	A4301, 25.09.2017.
<b>OIB studenta:</b>	39168964140
<b>Mentor:</b>	Prof.dr.sc. Robert Cupec
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	Boris Mandić
<b>Predsjednik Povjerenstva:</b>	Prof.dr.sc. Dražen Slišković
<b>Član Povjerenstva:</b>	Doc.dr.sc. Damir Filko
<b>Naslov završnog rada:</b>	Sekvencioner za pokretanje hidrauličke jedinice
<b>Znanstvena grana rada:</b>	<b>Automatizacija i robotika (zn. polje elektrotehnika)</b>
<b>Zadatak završnog rada</b>	Treba izraditi programski blok za automatsko pokretanje hidrauličke jedinice. Sustav se može sastojati od n pumpi te imati zahtjev za pokretanje m pumpi. Rješenje treba voditi računa o ispunjenosti uvjeta za uključivanje svake pojedine pumpe te ih uključivati u ovisnosti o broju radnih sati svake od njih. Po istom principu potrebno je uključiti rezervnu pumpu u slučaju iznenadnog gašenja neke od pumpi. Završni rad uključuje razvoj softvera i simulaciju u alatu Step7 i CoDeSys te vizualizaciju u alatu CoDeSys. ( Sumentor: Boris Mandić, Danieli-Systec d.o.o, Vinež 601, 52220 Labin)
<b>Prijedlog ocjene pismenog dijela ispita (završnog rada):</b>	Vrlo dobar (4)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 2 razina
<b>Datum prijedloga ocjene mentora:</b>	18.09.2018.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 02.10.2018.

**Ime i prezime studenta:**

Luka Strinić

**Studij:**

Preddiplomski stručni studij Elektrotehnika, smjer Automatika

**Mat. br. studenta, godina upisa:**

A4301, 25.09.2017.

**Ephorus podudaranje [%]:**

6%

Ovom izjavom izjavljujem da je rad pod nazivom: **Sekvencioner za pokretanje hidrauličke jedinice**

izrađen pod vodstvom mentora Prof.dr.sc. Robert Cupec

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

## Sadržaj

1. UVOD.....	1
1.1. Zadatak završnog rada.....	1
2. PROGRAMSKI BLOK ZA AUTOMATSKO POKRETANJE HIDRAULIČKE JEDINICE.....	2
3. PROGRAMABILNI LOGIČKI KONTROLERI (PLC).....	5
3.1. STEP 7.....	6
3.1.1. Statement list (STL).....	7
3.1.2. Function block diagram (FBD).....	7
3.1.3. Ladder logic (LAD).....	8
3.1.4. Vrste podataka.....	8
4. PROGRAMSKO RJEŠENJE ZA AUTOMATSKO POKRETANJE HIDRAULIČKE JEDINICE.....	10
4.1. Pokretanje hidrauličke pumpe.....	10
4.1.1. Automatsko pokretanje.....	11
4.1.2. Lokalno pokretanje pumpe.....	13
4.1.3. Pokretanje Stand-by pumpe.....	13
4.2. Zaustavljanje hidrauličke pumpe.....	15
4.3. Brojanje radnih sati pumpe.....	16
4.4. Određivanje stand-by pumpe.....	17
5. CODESYS.....	21
5.1. Programski jezici.....	21
5.2. Rješenje problema Sekvencionalnog pokretanja hidrauličke jedinice.....	21
5.3. CODESYS vizualizacija.....	22
6. ZAKLJUČAK.....	24
LITERATURA.....	25
SAŽETAK.....	26
ABSTRACT.....	27
ŽIVOTOPIS.....	28

## **1. UVOD**

U središtu ovog rada prikazan je programski blok za automatizaciju pokretanja hidrauličke jedinice. Automatizacija spada u proces kojim se nešto izrađuje ili pravi automatski. Nakon mehanizacije određenog zadatka slijedi proces automatizacije. Takvim načinom rada dovodimo do smanjenja potrebe ljudskog fizičkog rada, odlučivanja i opažanja. Današnjim radom i načinom funkcioniranja društva slobodno možemo reći da živimo u doba automatizacije, gdje većina sustava bez primijenjenog načela automatskog upravljanja nije u mogućnosti ispravno raditi ili uopće raditi. Automatsko upravljanje spada u inženjerske znanosti te predstavlja jedan od temelja funkcioniranja današnjeg društva.

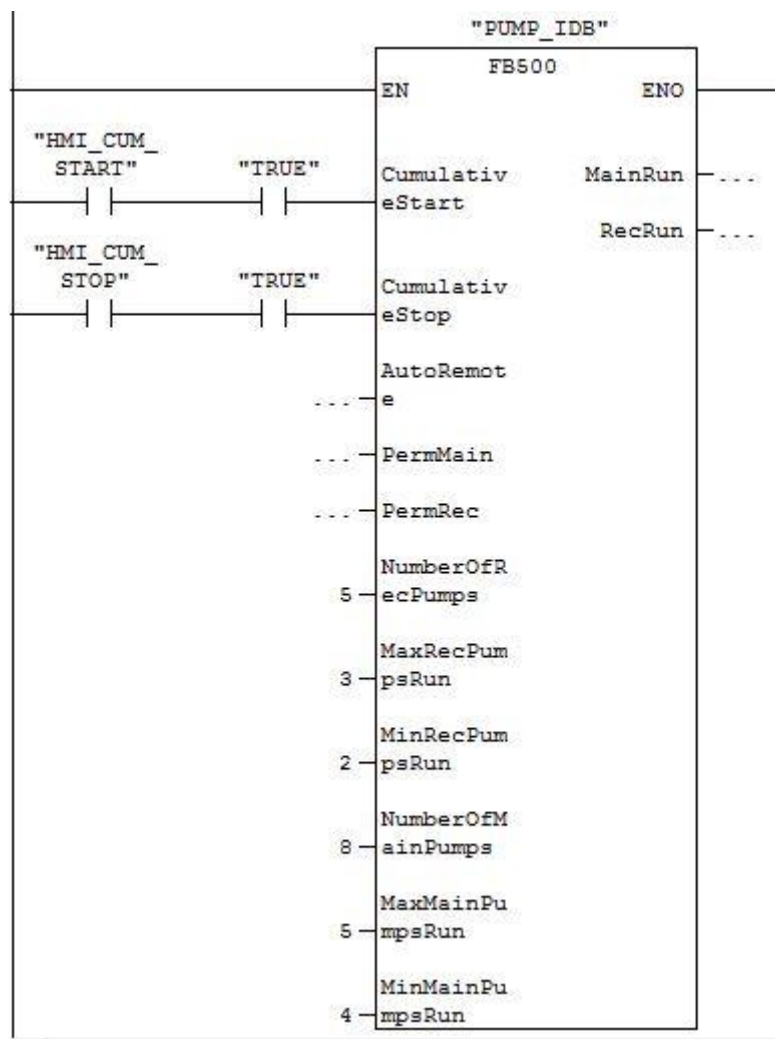
Prvi dio ovog rada predstavlja uvod s detaljnim opisom zadatka rada. To je prvo poglavlje od ukupno šest koji čine ovaj završni rad. U drugom dijelu opisan je blok za automatsko pokretanje hidrauličke jedinice. U trećem poglavlju upoznat će se s programabilnim logičkim kontrolerima ( PLC- ovima), o načinu njihovog rada, koja je njihova primjena te princip njihovog funkcioniranja. Također, upoznat će se s programom za njihovo programiranje STEP7. U četvrtom poglavlju prikazano je programsko rješenje te način kako voditi računa o ispunjenosti uvjeta za uključivanje svake pojedine pumpe. Uključivanje pumpi ovisit će o broju radnih sati iste, te je prikazan princip uključivanja rezervne pumpe u slučaju iznenadnog gašenja glavne pumpe. Peto poglavlje obuhvaća temu vizualizacije softverskog dijela rada. Upoznat će se s programskim alatom Codesys. Zadnje poglavlje obuhvaća zaključak rada.

### **1.1. Zadatak završnog rada**

Zadatak ovog završnog rada je izraditi programski blok za automatsko pokretanje hidrauličke jedinice. Sustav se sastoji od n recirkulacijskih pumpi te m distribucijskih pumpi. Programskim blokom treba provjeravati ispunjenost uvjeta za uključivanje svake pojedine pumpe, te realizirati njihovo uključivanje ovisno o broju radnih sati svake pojedine pumpe. Također, po istom principu potrebno je uključiti rezervnu pumpu u slučaju iznenadnog gašenja neke od pumpi. Zadatak završnog rada uključuje i razvoj programa i simulaciju u alatu Step7 i CoDeSys te vizualizaciju u alatu CoDeSys.

## 2. PROGRAMSKI BLOK ZA AUTOMATSKO POKRETANJE HIDRAULIČKE JEDINICE

Programski blok za automatsko pokretanje hidrauličke jedinice baziran je na sekvencijalnom upravljanju. Jedan od najčešćih načina upravljanja procesa je upravo sekvencijalno upravljanje. Sekvencijalnim upravljanjem izvršnim članovima u procesu zadajemo određeni niz naredbi, npr. ( kreni/stani, uključi/isključi i sl. ), pri čemu svaki proces traje unaprijed određeni vremenski interval. Na primjer, pri pokretanju hidrauličke jedinice, treba voditi računa da se pumpe pokreću slijedno odnosno jedna po jedna kako ne bi došlo do preopterećenja mreže odnosno ispada mreže.

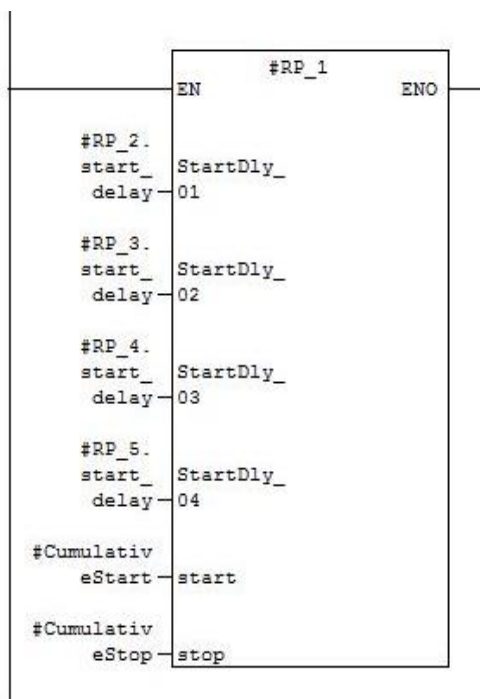


Slika 2.1. Programski blok za automatsko pokretanje hidrauličke jedinice

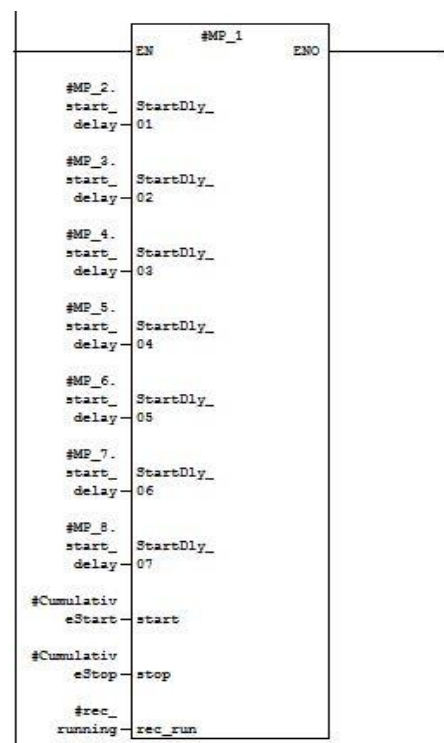
Na slici 2.1. prikazan je funkcijski blok za automatsko pokretanje hidrauličke jedinice. Funkcijski blok posjeduje vlastiti podatkovni blok (PUMP\_IDB) koji u svojoj memoriji sadrži strukturu funkcijskog bloka i sve podatke funkcijskog bloka. Na ulazu funkcijskog bloka prikazane su osnovne naredbe i informacije. „CumulativeStart“ i „CumulativeStop“ služe za pokretanje i

zaustavljanje hidrauličke jedinice. „PermMain“ i „PermRec“ prikazuju imaju li distribucijske i recirkulacijske pumpe ispunjene uvijete za rad. „AutoRemote“ označava upravlja li se hidrauličkom jedinicom automatski preko Human-machine interface ( HMI ) ili preko lokalne kontrolne stanice. „NumberOfMainPumps“ „NumberOfRecPumps“ daju informacije od koliko se distribucijskih i recirkulacijskih pumpi sastoji hidraulička jedinica. „MaxMainPumps“, „MinMainPumps“, „MaxRecPumps“ i „MinRecPumps“ prikazuju koliko maksimalno i minimalno distribucijskih i recirkulacijskih pumpi mora raditi da bi hidraulička jedinica radila. „MainRun“ i „RecRun“ na izlazu funkcijskog bloka daju informacije o tome radi li hidraulička jedinica. Varijable „HMI\_CUM\_START“ i „HMI\_CUM\_STOP“ predstavljaju HMI komande pomoću kojih operater pokreće ili zaustavlja hidrauličku jedinicu.

Unutar funkcijskog bloka FB500 napisana je ostala logika upravljanja hidrauličkom jedinicom. Recirkulacijske i distribucijske pumpe posjeduju posebne funkcijske blokove u kojima je napisano programsko rješenje za pokretanje, zaustavljanje, brojanje radnih sati te ostalo. Svaki blok je identičan te koliko ima recirkulacijskih i distribucijskih pumpi toliko se puta pozove blok za pumpu, te im se pridruže drugačiji podatkovni blokovi. Prikaz funkcijskog bloka za recirkulacijske i distribucijske pumpe prikazan je na slikama 2.2. i 2.3.



Slika 2.2. Funkcijski blok za recirkulacijske pumpe



Slika 2.3. Funkcijski blok za distribucijske pumpe



Na ulazu svakog funkcijskog bloka za distribucijske i recirkulacijske pumpe potrebno je dodati komande za start i stop odnosno varijable „CumulativStart“ i „CumulativStop“, također dodaju se varijable „start\_delay“ pomoću kojih se sprječava pokretanje više pumpi u isto vrijeme. Svaka pumpa koja dobije zahtjev za automatsko pokretanje, se pokrene i aktivira OFF Delay Timer koji postavlja varijablu „start\_delay“ u 1 na određeno vrijeme. Dok je varijabla „start\_delay“ aktivna ni jedna druga pumpa nije u mogućnosti krenuti jer ne ispunjava sve uvijete za pokretane pumpe prema slici ( 4.1. ).

Svaki funkcijski blok za distribucijske ili recirkulacijske pumpe na izlazu daje podatak o tome radi li pumpa ili ne. Također daje radno vrijeme te iste pumpe.

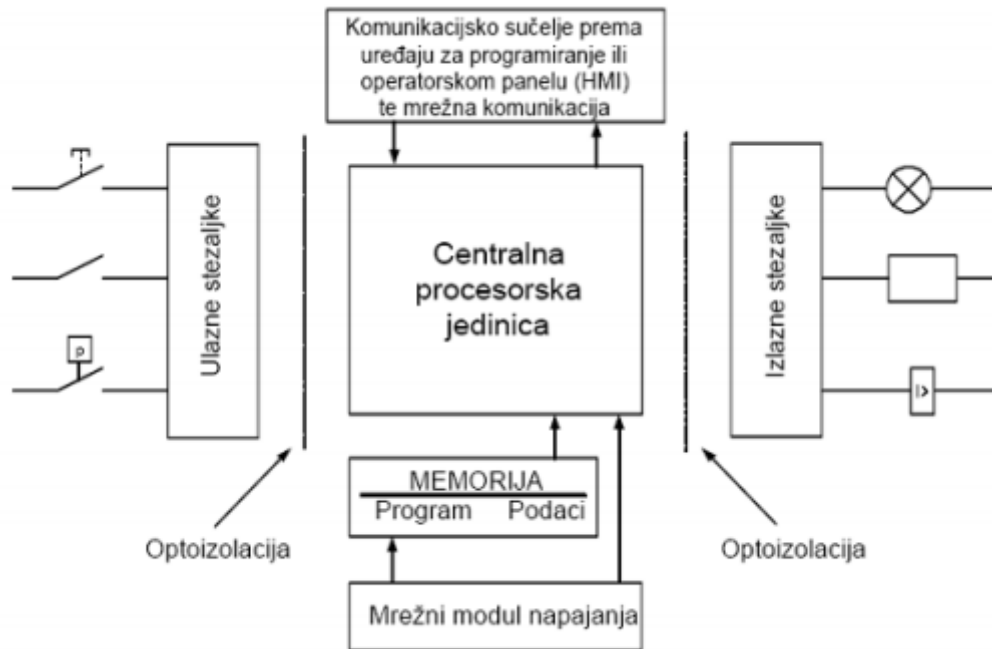
### 3. PROGRAMABILNI LOGIČKI KONTROLERI (PLC)

Sve do 60-ih godina prošlog stoljeća industrijskim sustavima se upravljalo pomoću sklopnika i releja odnosno pomoću relejne logike. Relejna logika predstavlja niz komponenata kao što su releji, sklopnici te ostale komponente koje su bile spojene žicama te se na taj način vršila logika upravljanja sustavom. Najveći problem kod ovakvog načina upravljanja sustavima, posebno složenim sustavima, je taj što ako se dogodi pogreška, želimo promijeniti logiku upravljanja ili dodati nove komponente u sustav upravljanja, to moramo učiniti prespajanjem žica. Zbog toga i zbog olakšanja upravljanja industrijskim sustavima uvodi se središnje računalo sustava odnosno PLC. PLC-ovi su snažna i modularna računala namijenjena za rad u industriji. Predstavljaju središnja računala u industrijskoj automatizaciji. Njihova funkcija u procesu automatizacije je da primaju podatke od mjernih članova iz procesa, s obzirom na primljene podatke od mjernih članova i s obzirom na softversko rješenje procesa koje je isprogramirano u PLC-u prosljeđuju unaprijed određene naredbe ili veličine izvršnim članovima, te na taj način upravljaju radom nekih uređaja poput motora, frekvencijskog pretvarača, hidrauličkih pumpi i sl.



Slika 3.1. Princip rada PLC-a

Najveća prednost PLC je njegova modularnost, odnosno da pomoću različitih modula ( signalnih modula, funkcijskih modula, procesorskih jedinica itd.) možemo konfigurirati PLC baš onakvih karakteristika kakve su nam potrebne za upravljanje određenim sustavom.



Slika 3.2. Osnovne cjeline PLC uređaja

### 3.1. STEP 7

STEP 7 je Siemensov program iz porodice SIMATIC S7 pomoću kojeg se programiraju PLC-ovi. Program se temelji na IEC 61131-3 standardu. Detaljnije podatke o programu moguće je pronaći u priručniku [1].

Programiranje PLC-ova pomoću programa STEP 7 moguće je u više programskih jezika.

Osnovni programski jezici u STEP 7 su :

- STL ( statement list)
- FBD (function block diagram)
- LAD ( ladder logic)

### 3.1.1. Statement list (STL)

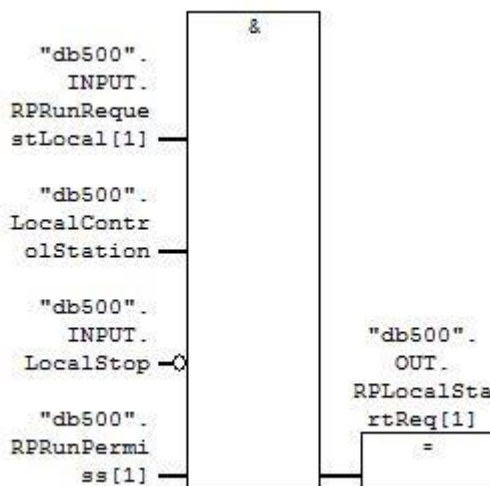
STL je programski jezik najbliži programerima pomoću kojeg jednostavnim naredbama opisujemo operacije koje PLC mora izvršiti. Ovaj programski jezik ima modularnu građu gdje nam svaki redak predstavlja jednu naredbu. Popis naredbi definiran je IEC 61131-3 standardom.

```
Network 1 : Title:
A      "db500".INPUT.RPRunRequestAutomatic[1]
A      "db500".RPRunPermiss[1]
AN     "db500".PauzaStart[1]
AN     "db500".LocalControlStation
=      L      0.0
A      L      0.0
BLD    102
=      "db500".OUT.RPAutomaticStartReq[1]
A      L      0.0
FP     "db500".AUX_ONS[0]
S      "db500".PauzaStart[1]
```

Slika 3.3. Prikaz funkcije napisan STL programskim jezikom

### 3.1.2. Function block diagram (FBD)

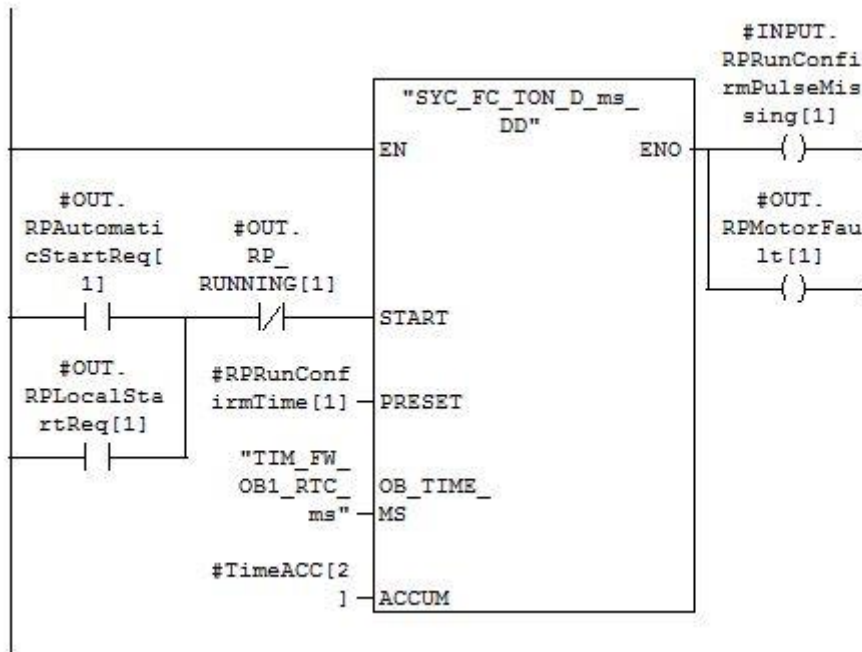
FBD spada u grafički jezik za programiranje PLC-ova. Programiranje se vrši slaganjem i povezivanjem funkcijskih blokova kako bi dobili isti učinak kao s LAD jezikom. Svaki funkcijski blok nam predstavlja naredbu koja se sastoji od određenog broja ulaza i izlaza.



Slika 3.4. Funkcija napisan FBD programskim jezikom

### 3.1.3. Ladder logic ( LAD)

Ladder logic je najpopularniji jezik za programiranje PLC-ova. Vrlo je jednostavan za shvaćanje, u početku je korišten za programiranje jednostavnih sustava ali kako su s vremenom sustavi postali sve kompleksniji tako su se počeli programirati i sve složeniji sustavi. Program se izvršava s lijevo na desno sve do kraja nakon toga proces počinje ispočetka.



Slika 3.5. Funkcija napisana LAD programskim jezikom

### 3.1.4. Vrste podataka

STEP 7 program sadrži određene tipove podataka koje smijemo koristiti. Prilikom programiranja moramo poštovati format unosa podataka, u protivnom program neće raditi. Svaka vrsta podatka zauzima definiranu količinu memorije koja je izražena u bitovima.

Vrsta Podatka i opis	Veličina koju podatak zauzima u bitovima	Dozvoljeni format
BOOL (bit)	1	Boolean
BYTE (byte)	8	Heksadecimalni broj
WORD (riječ)	16	BCD,binarni broj,heksadecimalni broj,ne pridruženi dekadski broj
DWORD (dupla riječ)	32	Heksadecimalni broj,binarni broj,ne pridruženi broj
INT (integer)	16	Dekadski broj
DINT (dupli integer)	32	Dekadski broj
REAL (broj s pomičnom točkom)	32	IEEE broj s pomičnom točkom
S5TIME(SIMATIC vrijeme)	16	S7 vrijeme u koracima od 10ms
TIME (IEC vrijeme)	32	IEC vrijeme u koracima od 1ms
DATE (IEC datum)	16	IEC vrijeme u koracima od 1 dan
TIME_OF_DAY (vrijeme)	32	Vrijeme u koracima od 1 ms
CHAR (znak)	8	ASCII znak

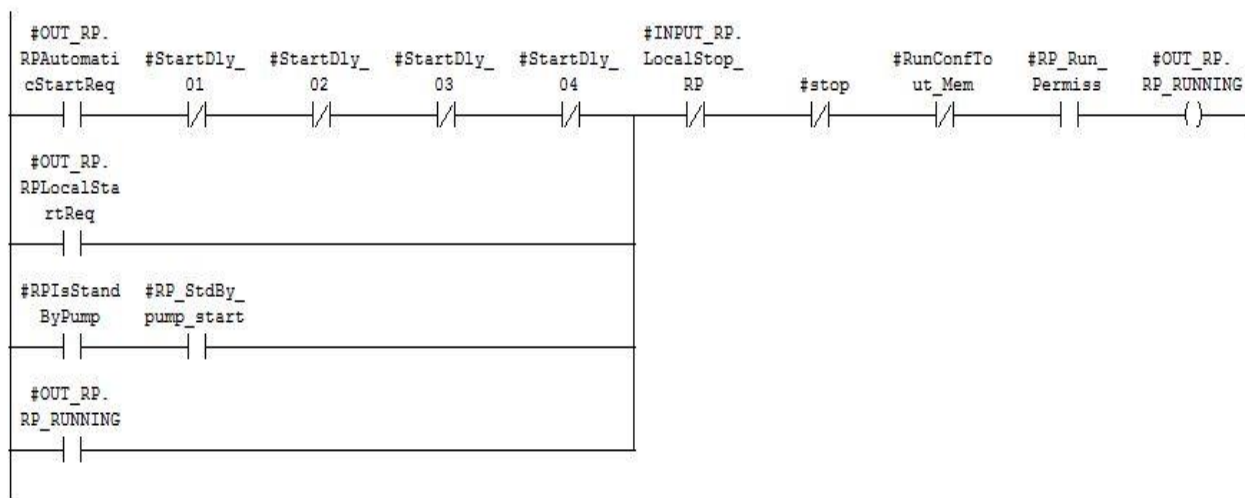
Tablica 3.1 Osnovne vrste podataka

## 4. PROGRAMSKO RJEŠENJE ZA AUTOMATSKO POKRETANJE HIDRAULIČKE JEDINICE

Programsko rješenje za automatsko pokretanje hidrauličke jedinice je realizirano u razvojnom okruženju STEP7 pomoću LAD programskog jezika. Hidraulička jedinica se smatra pokrenuta ako u njoj radi određen broj recirkulacijskih i distribucijskih pumpi.

### 4.1. Pokretanje hidrauličke pumpe

Pumpu je moguće pokrenuti u tri slučaja. Prvi način pokretanja pumpe je automatski način pokretanja, drugi način pokretanja je lokalni preko lokalne kontrolne stanice i treći način pokretanja je stand-by pokretanje odnosno ako je ta pumpa trenutno stand-by pumpa i došao je zahtjev za pokretanje stand-by pumpe. Pumpa će se pokrenuti ako je došao jedan od gore navedenih zahtjeva i ako ima sve uvijete za rad. Način pokretanja pojedine pumpe u hidrauličkoj jedinici prikazan je na slici 4.1. i on je jednak za sve pumpe u hidrauličkoj jedinici.



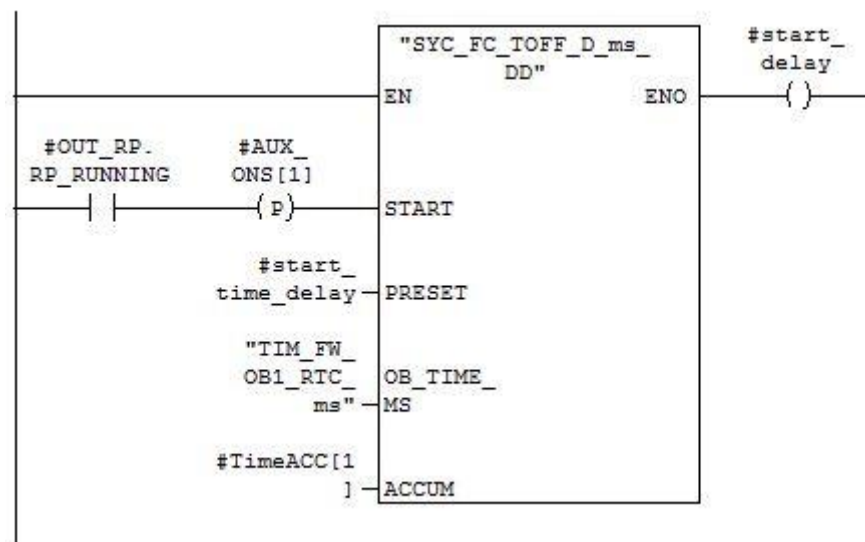
Slika 4.1. Način pokretanja Pumpe

Signali koji utječu na pokretanje i zaustavljanje pumpe dolaze s lokalne kontrolne stanice te iz sustavnih blokova nakon njihova izvršenja. Signali „OUT\_RP.RPLocalStartReq“, „stop“ i „INPUT\_RP.LocalStop\_RP“ su signali koji dolaze s lokalne kontrolne stanice, njima se upravlja pomoću tipkala ili sklopki, ovisno o izvedbi kontrolne stanice. Signal „OUT\_RP.RP\_RUNNING“ nakon ispunjavanja zadanih uvjeta, odnosno logike koja je prikazana na slici 4.1, pokreće pumpu koja je spojena na stvarnu adresu varijable. Preostali signali sa slike 4.1. dolaze iz sustava.

Naziv varijable	Značenje varijable
#OUT_RP.RPAutomaticStartReq	Zahtjev za automatsko pokretanje pumpe
#Start_Dly	Vremenska odgoda pokretanja pumpe
INPUT_RP.LocalStopRP	Lokalno zaustavljanje pumpe
#stop	Zahtjev za zaustavljanje hidrauličke jedinice
#RunConfTout_Mem	Signal nereagiranja pumpe na zahtjev za start
#RP_Run_Permiss	Uvjet za rad pumpe
#OUT_RP.RP_RUNNING	Recirkulacijska pumpa
#OUT_RP.RPLocalStartReq	Lokalno pokretanje pumpe
#RpIsStandByPump	Recirkulacijska pumpa je stand-by pumpa
#RP_StdBy_pump_start	Recirkulacijska pumpa ima zahtjev da se uključi kao stand-by pumpa

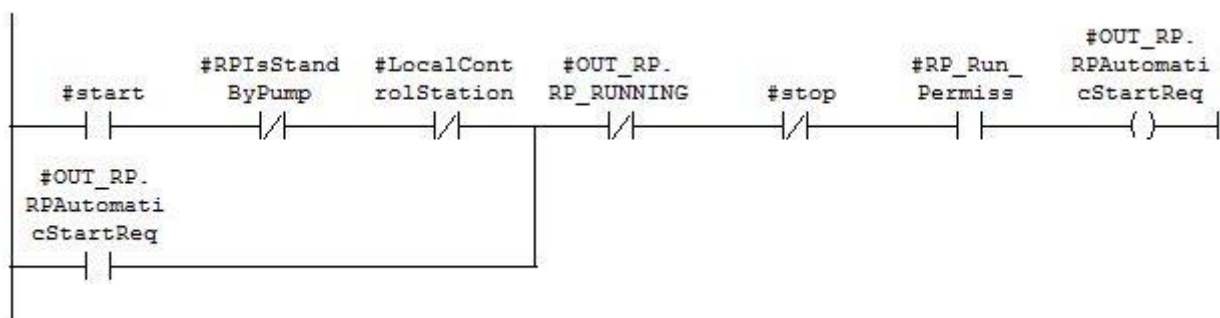
Tablica 4.1. Objašnjenje varijabli korištenih u programu

#### 4.1.1. Automatsko pokretanje



Slika 4.2. Vremenski interval za pokretanje pumpi





Slika 4.3. Zahtjev za automatsko pokretanje pumpe

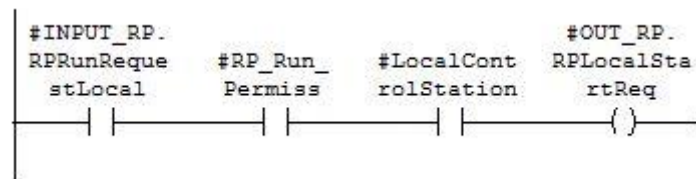
Na slici 4.3. prikazano je programsko rješenje slanja zahtjeva za pokretanje pumpe u automatskom načinu rada. Kod automatskog načina rada pumpa će se pokrenuti nakon isteka vremenskog intervala koji je definiran pomoću tajmera prikazanog na slici 4.2., uz uvjet da pumpa ima sve potrebne uvijete za pokretanje. Kada pumpi dođe zahtjev za automatsko pokretanje, pumpa koja je dobila zahtjev pokreće timer koji aktivira varijablu „start\_delay“ te na taj način sprječava pokretanje bilo koje druge pumpe koja ima zahtjev za pokretanje sve dok varijabla „start\_delay“ ne poprimi vrijednost nula.

Naziv varijable	Značenje varijable
#start	Zahtjev za pokretanje hidrauličke jedinice
#RpIsStandByPump	Recirkulacijska pumpa je stand-by pumpa
#LocalControlStation	Lokalna kontrola stanica (1= aktivna , 0= neaktivna)
#OUT_RP.RP_RUNNING	Recirkulacijska pumpa
#stop	Zahtjev za zaustavljanje hidrauličke jedinice
#RP_Run_Permiss	Uvjet za rad pumpe
#OUT_RP.RPAutomati cStartReq	Zahtjev za automatsko pokretanje pumpe
#start_delay	Odgoda uključivanja ostalih pumpi
#AUX_ONS	Pomoćna varijabla za jedan impulsni signal
#start_time_delay	Vrijeme odgode uključivanja pumpi
#TIM_FW_OB1_RTC_ms	Vrijeme rada operacijskog bloka
TimeACC	Pomoćni akumulator u koje se sprema vrijeme rada operacijskog bloka.

Tablica 4.2. Objašnjenje varijabli korištenih u programu

### 4.1.2. Lokalno pokretanje pumpe

Lokalno pokretanje pumpe se izvodi preko lokalne kontrolne stanice. U lokalnom načinu pokretanja nije definiran vremenski interval te je teoretski moguće sve pumpe pokrenuti istovremeno. Također u lokalnom načinu rada ako bi došlo do ispada neke od pumpi stand-by pumpa se ne bi pokrenula. Da bi se pumpa pokrenula u lokalnom načinu rada, lokalna kontrolna stanica mora biti aktivirana, pumpa mora imati sve potrebne uvijete za rad. Ukoliko su svi ti uvjeti ispunjeni aktivira se varijabla „OUT\_RP.RPLocalStartReq“ te se prema logici prikazanoj na slici (4.1.) pumpa pokreće.



Slika 4.4. Slanje lokalnog zahtjeva za pokretanje pumpe

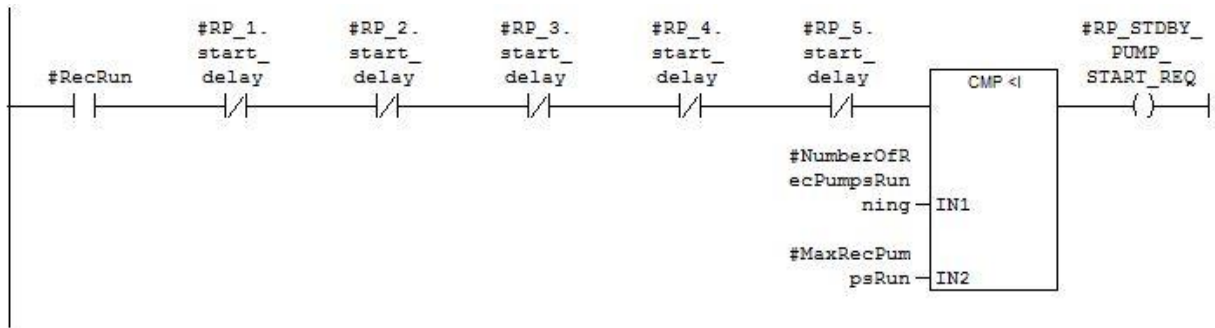
Naziv varijable	Značenje varijable
#INPUT_RP.RPRunRequestLocal	Zahtjev za lokalno pokretanje pumpe
#RP_Run_Permiss	Uvjet za rad pumpe
#LocalControlStation	Lokalna kontrola stanica (1= aktivna , 0= neaktivna)
#OUT_RP.RPLocalStartReq	Lokalno pokretanje pumpe

Tablica 4.3. Objašnjenje varijabli korištenih u programu

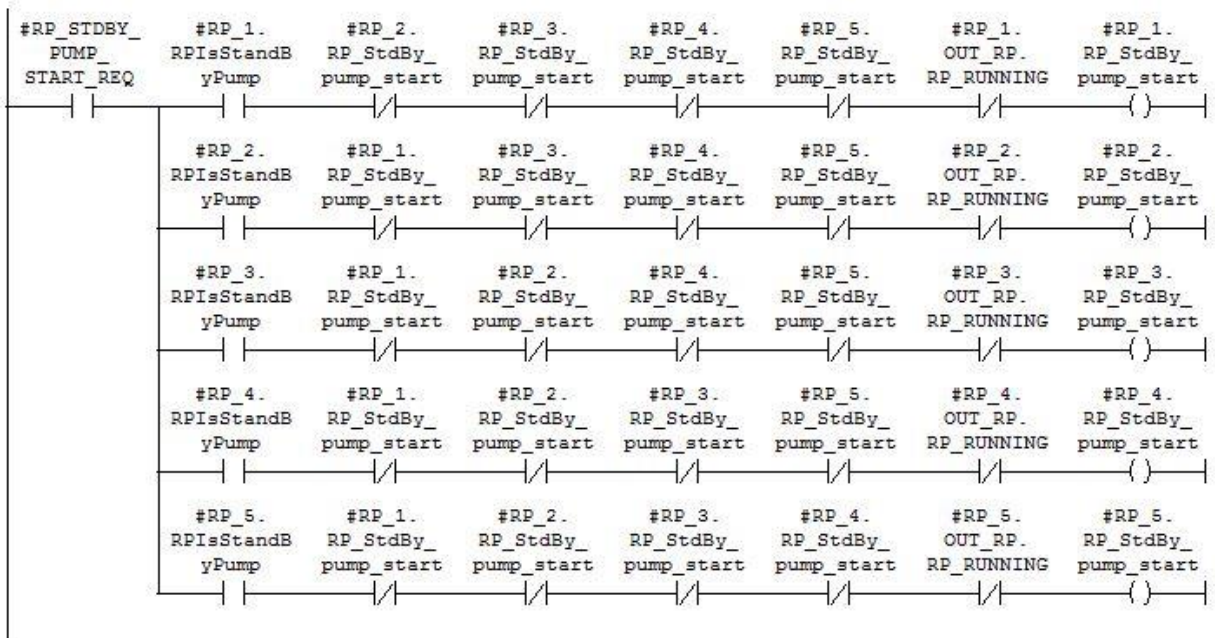
### 4.1.3. Pokretanje Stand-by pumpe

Stand-by pumpa će se pokrenuti kada dođe zahtjev za pokretanje iste. Odnosno ako se dogodi prekid rada jedne od pumpi u automatskom načinu rada. Zahtjev za pokretanje stand-by pumpe bit će poslan u slučaju da je pumpa radila i u jednom trenutku prestala s radom ili da se pri pokretanju hidrauličke jedinice u automatskom načinu rada nije pokrenula nakon isteka vremenskog intervala za pokretanje. Na slici 4.5. prikazano je programsko rješenje slanja zahtjeva za pokretanje jedne od stand-by recirkulacijskih pumpi. Zahtjev za pokretanje stand-by pumpe će doći ako

recirkulacijska jedinica radi, ni jedna od recirkulacijskih pumpi nije u fazi pokretanja te je broj pumpi koje rade manji od unaprijed definiranog maksimalnog broja radnih pumpi.



Slika 4.5. Zahtjev za pokretanje stand-by pumpe



Slika 4.6. Pokretanje stand-by pumpe

Naziv varijable	Značenje varijable
#RecRun	Recirkulacijska jedinica( 1=radi, 0= ne radi)
#RP_start_delay	Odgoda pokretanja pumpe
#NumberOfRecPumpsRunning	Broj recirkulacijskih pumpi koje rade

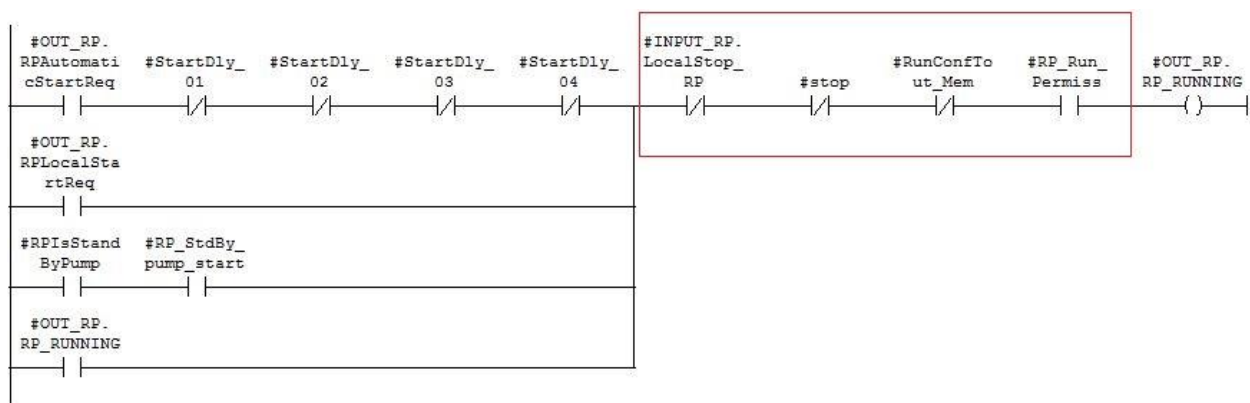
#MaxRecPumpsRun	Broj recirkulacijskih pumpi koje moraju raditi da bi se recirkulacijska jedinica smatrala uključenom
#RP_STDBY_PUMP_START_REQ	Zahtjev za pokretanje stand-by pumpe
#RpIsStandByPump	Recirkulacijska pumpa je stand-by pumpa
#OUT_RP.RP_RUNNING	Recirkulacijska pumpa
#RP_StdBy_pump_start	Recirkulacijska pumpa ima zahtjev da se uključi kao stand-by pumpa

Tablica 4.4. Objašnjenje varijabli korištenih u programu

Slika 4.6. prikazuje programsko rješenje za pokretanje recirkulacijske stand-by pumpe. Stand-by pumpa će se pokrenuti ako ispuni sljedeće uvijete. Da bi se pokrenula jedna od stand-by pumpi, mora doći zahtjev za pokretanje stand-by pumpe, pumpa mora biti postavljena kao stand-by pumpa te ni jedna ostala stand-by pumpa ne smije imati zahtjev za start. Na isti princip riješeno je pokretanje distribucijskih stand-by pumpi.

## 4.2. Zaustavljanje hidrauličke pumpe

Zaustavljanje hidrauličke pumpe moguće je obaviti na više načina. Jedan od načina je pritiskom komande „CumulativStop“ preko koje se zaustavlja kompletna hidraulička jedinica.



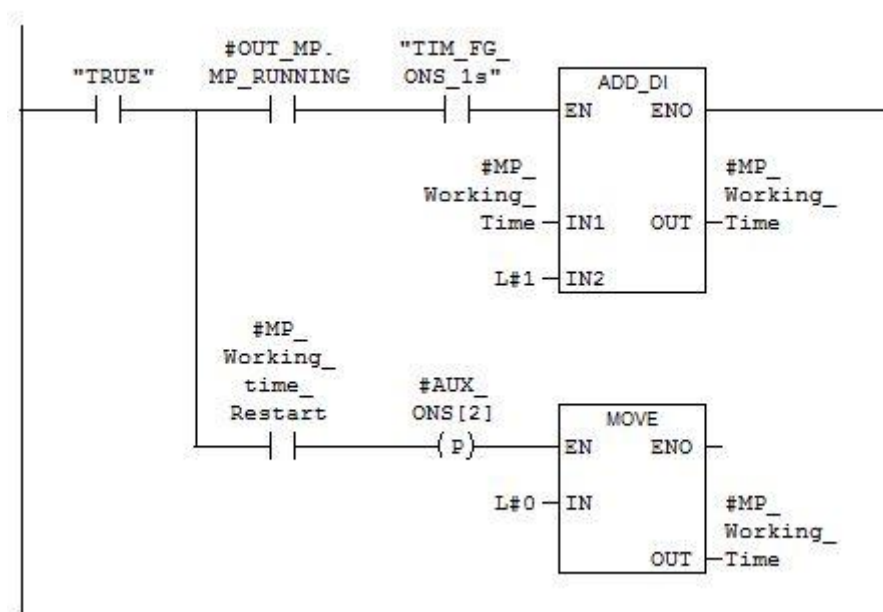
Slika 4.7. Zaustavljanje hidrauličke pumpe

Hidraulička pumpa će također prekinuti s radom ako izgubi ili dobije jedan od sljedećih uvjeta koji su označeni na slici 4.7. Varijabla „INPUT\_RP.LocalStopRP“ služi za lokalno zaustavljanje

pumpe, uz uvjet da je LCS aktiviran, inače lokalno zaustavljanje nije moguće napraviti. Do lokalnog zaustavljanja pumpe će doći ukoliko operater na LCS-u pritisne tipkalo za zaustavljanje pumpe te tako aktivira varijablu „INPUT\_RP.LocalStopRP“. Varijabla „RunConfTout\_Mem“ zaustavlja pumpu ako poprimi vrijednost 1. Varijabla će dobiti vrijednost 1 ako iz LCS-a ili HMI-a dođe zahtjev za lokalno ili automatsko pokretanje pumpe te pumpa nakon određenog vremenskog intervala nije poslala povratni signal da radi. Ako tijekom rada pumpa izgubi neki od uvijeta za rad „RP\_Run\_Permiss“, koji se provjeravaju u funkcijskom bloku za svaku pojedinu pumpu, pumpa će također prekinuti s radom.

### 4.3. Brojanje radnih sati pumpe

Programsko rješenje brojanja radnih sati pumpe i resetiranje radnih sati prikazano je na slici 4.8. Funkcija ADD\_DI služi za zbrajanje cijelih brojeva dvostruke duljine ako se na ulazu funkcije dogodi rastući brid i tu vrijednost sprema na izlaz funkcije. „TIM\_FG\_ONS\_1s“ koji se nalazi na ulazu funkcije ADD\_DI daje rastući brid svake sekunde te povećava vrijednost „MP\_Working\_Time“ za 1 svake sekunde i tu vrijednost sprema u „MP\_Working\_Time“ koja predstavlja radno vrijeme pumpe. Svaka pumpa ima mogućnost resetiranja broja radnih sati. Resetiranje radnih sati izvršit će se ako dođe zahtjev za resetiranje. Ako dođe zahtjev za resetiranje broja radnih sati pumpe, pomoću naredbe MOVE prebacuje se 0 u radno vrijeme pumpe „RPWorkingSeconds“.



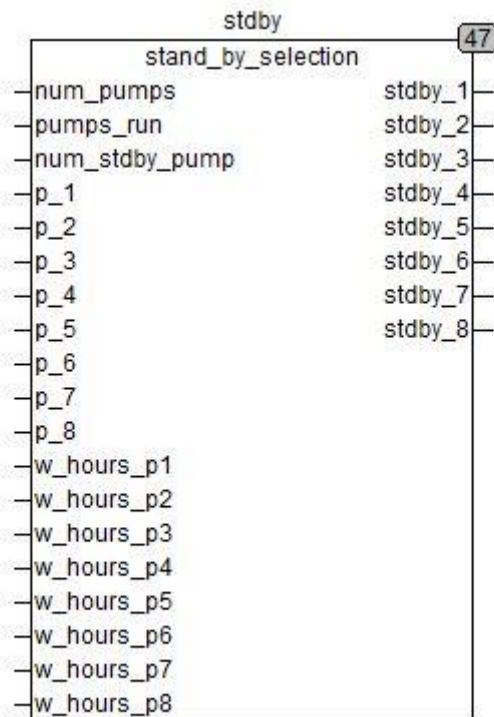
Slika 4.8. Brojanje radnih sati i resetiranje radnih sati pumpe.

Naziv varijable	Značenje varijable
„TRUE“	Varijbla koja je uvijek aktivna
#OUT_RP.RP_RUNNING	Recirkulacijska pumpa
„TIM_FG_ONS_1s“	Varijbla koja svake sekunde postane aktivna
#MP_Working_Time	Radno vrijeme pumpe
#MP_Working_time_Restart	Resetiranje radnog vremena pumpe
#AUX_ONS	Pomoćna varijbla za jedan impulsni signal

Tablica 4.5. Objašnjenje varijabli korištenih u programu

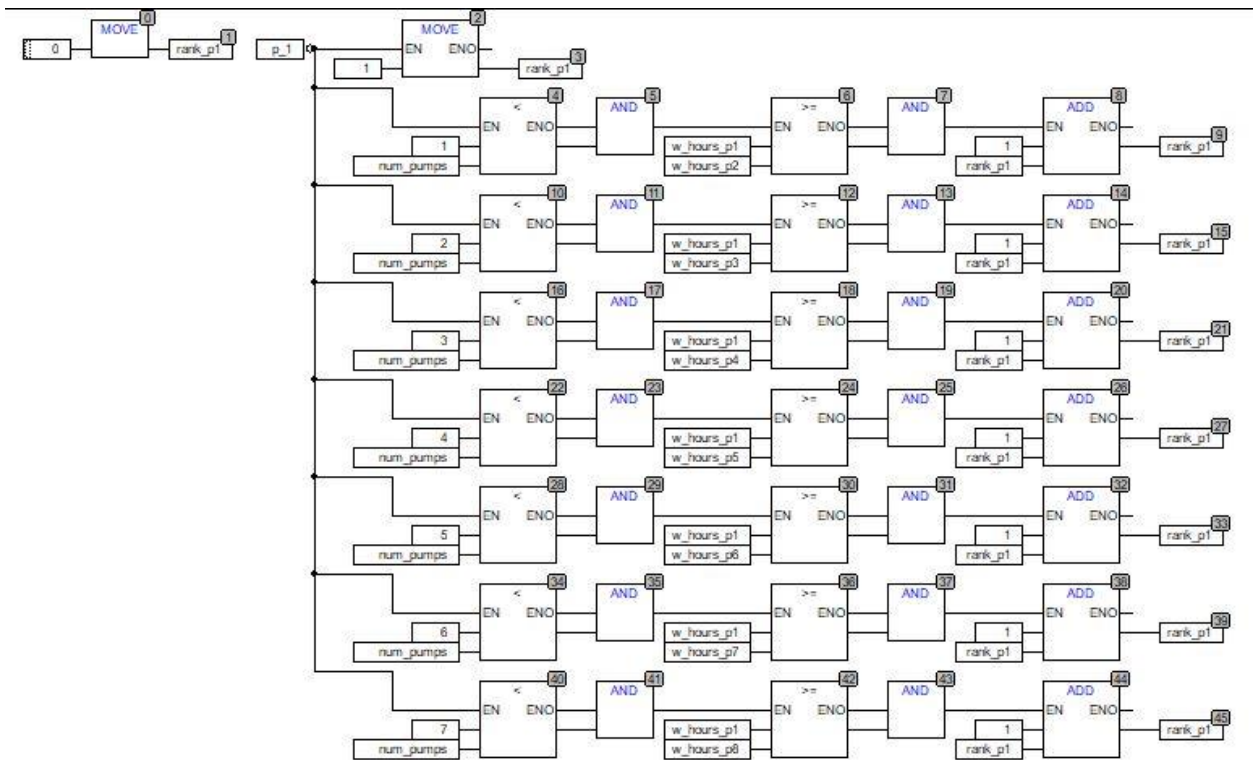
#### 4.4. Određivanje stand-by pumpe

Na slici 4.9. prikazan je izgled bloka funkcije koji služi za određivanje stand-by pumpi. Na ulazu u blok nalaze se varijable. Varijabli „num\_pumps“ pridružuje se broj pumpi od kojih se sastoji hidraulička jedinica. U varijablu „Pumps\_run“ pridružuje se vrijednost broja pumpi koje trenutno rade, a preko varijable „num\_stdby\_pump“ zadajemo koliki će biti broj stand-by pumpi u hidrauličkoj jedinici. Varijablami „p\_1“ do „p\_8“ pridružujemo stanje pumpi odnosno radi li pumpa ili ne, a varijablami „w\_hours\_p1“ do „w\_hours\_p8“ pridružujemo radna vremena pumpi. Na izlazu bloka funkcije dobivamo informaciju koja je pumpa postavljena kao stand-by pumpa.



Slika 4.9. Izgled bloka za određivanje stand-by pumpi

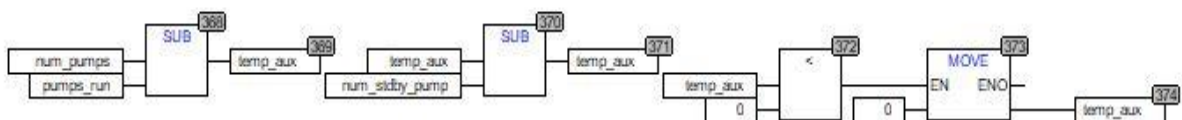
Stand-by pumpa se određuje tako što se uspoređuje radno vrijeme jedne pumpe s radnim vremenima svih ostalih pumpi. Uz uvjet da ta pumpa trenutno ne radi. Ako je radno vrijeme jedne pumpe veće od radnog vremena druge pumpe, toj pumpi se povećava rang za 1. Prije svakog pokretanja hidrauličke jedinice svi rangovi pumpe se stavljaju prvo u 0 i zatim se vrši usporedba sati i dodjeljivanje ranga za svaku pumpu.



Slika 4.10. Usporedba i dodjeljivanje ranga pumpe

Slika 4.10. prikazuje način usporedbe radnog vremena jedne pumpe sa svim ostalima te način dodjeljivanja ranga za tu istu pumpu.

Koliko pumpi treba biti stand-by i koliko stand-by pumpi treba pokrenuti ako dođe do prestanka rada glavnih pumpi određuje se tako što se oduzima ukupan broj pumpi koje se nalaze u hidrauličkoj jedinici s brojem pumpi koje trenutno rade. Ta vrijednost se sprema u pomoćnu varijablu „temp\_aux“, zatim se od varijable „temp\_aux“ oduzima broj stand-by pumpi i ta vrijednost se opet sprema u pomoćnu varijablu „temp\_aux“.

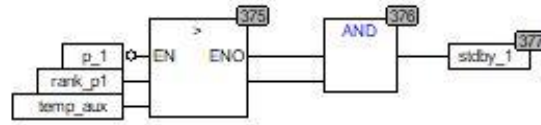


Slika 4.11. Izračun vrijednost varijable „temp\_aux“

Na slici 4.11. prikazan je način izračuna varijable „temp\_aux“.



Pomoću vrijednosti varijable „temp\_aux“ određuje se koje će pumpe biti stand-by pumpe. Svaka pumpa koja ima rang veći od vrijednosti varijable „temp\_aux“ postavlja se za stand-by pumpu. Način usporedbe prikazan je na slici 4.12.



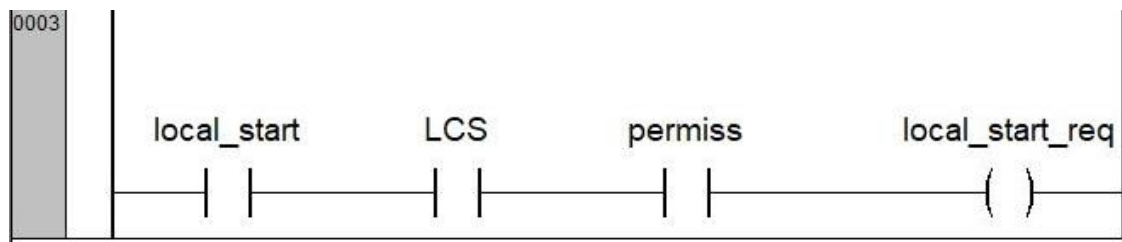
Slika 4.12. Postavljanje pumpe kao stand-by pumpe

## 5. CODESYS

CODESYS je programski alat koji služi za razvoj programskih rješenja. Omogućava nam programiranje PLC-ova i stvaranje vizualizacije sustava kojima upravlja PLC. Program je razvila njemačka kompanija Smart System Solutions. Temelji se na IEC 61131-3 standardu. Licence CODESYS-a su besplatne te ih je moguće legalno preuzeti na računalo. Dodatne informacije o programu je moguće pronaći u priručniku. [2]

### 5.1. Programski jezici

Kao i Simatic STEP 7, CODESYS se također temelji na IEC 61131-3 standardu, te omogućava programiranje u svim programskim jezicima koje propisuje gore navedeni standard, kao što su : Ladder Diagram (LD), The Continuous Function Chart Editor (CFC), Instruction List (IL), Structured Text (ST), Sequential Function Chart (SFC) i Function Block Diagram (FBD).

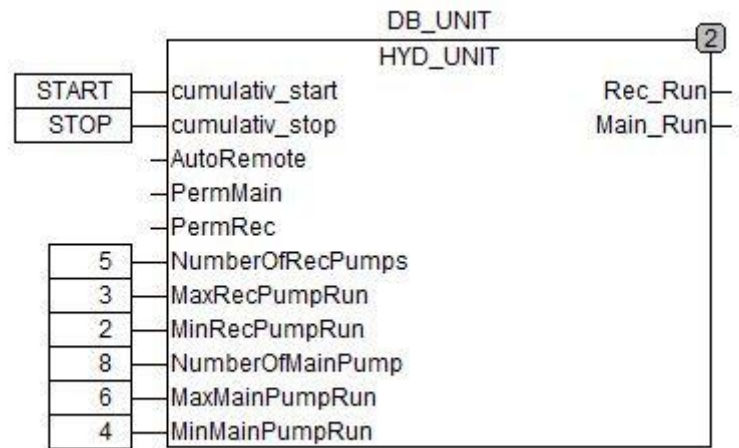


Slika 5.1. Ladder Diagram (LD)

Na slici 5.1. prikazan je izgled Ladder Diagram (LD) u CODESYS programskom alatu. Razlika između SIMATIC STEP 7 i CODESYS programskih jezika je vrlo mala.

### 5.2. Rješenje problema Sekvencionalnog pokretanja hidrauličke jedinice

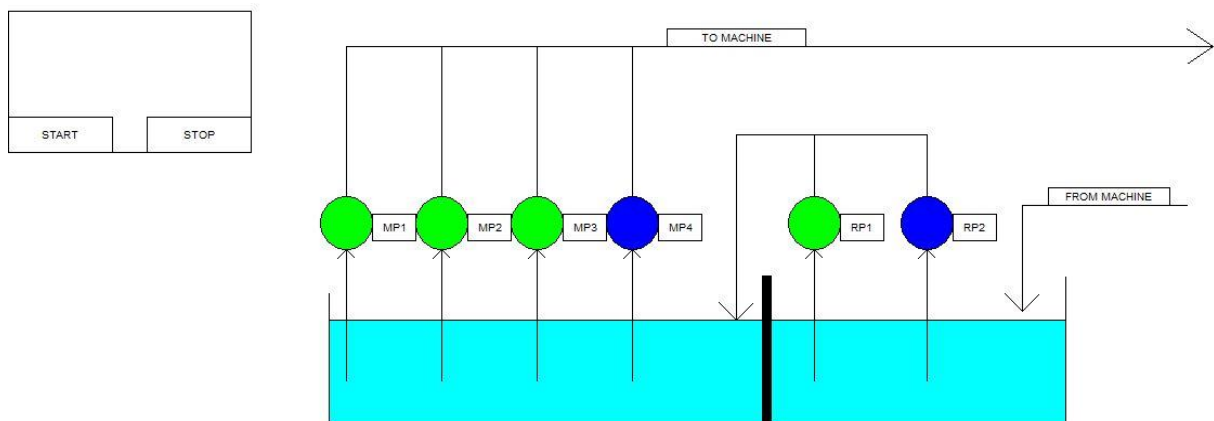
Programsko rješenje sekvencionalnog pokretanja hidrauličke jedinice u CODESYS-u pisano je programskim jezikom The Continuous Function Chart Editor (CFC). Programsko rješenje u CODESYS programskom alatu temelji se na logici programskog rješenja pisanog u SIMATIC STEP7 te zbog toga detaljna programska rješenja nisu prikazana. Na slici 5.2. prikazan je izgled glavnog bloka za automatsko pokretanje hidrauličke jedinice u CODESYS programskom alatu.



Slika 5.2. Izgled programskog bloka za automatsko pokretanje hidrauličke jedinice u CODESYS programskom alatu

### 5.3. CODESYS vizualizacija

Jedna od prednosti CODESYS programskog alata naspram SIMATIC STEP7 programskog alata je ta što CODESYS omogućava vizualizaciju programskog rješenja.



Slika 5.3. Vizualizacija u CODESYS programskom alatu

Slika 5.3 prikazuje vizualizaciju programskog rješenja. Iz slike je vidljivo da se hidraulička jedinica sastoji od 6 pumpi i dva odvojena spremnika. Manji spremnik koji sadrži tekućinu koja dolazi iz distribucijskog sustava. Dvije recirkulacijske pumpe od koji je jedna stand-by pumpa označena plavom bojom i jedna koja radi označena zelenom bojom. Recirkulacijske pumpe služe za recirkulaciju odnosno hlađenje i pročišćavanje tekućine, te takvu ohlađenu i pročišćenu

tekućinu pumpaju u veći spremnik iz kojega četiri distribucijske pumpe tu tekućinu dalje pumpaju prema strojevima odnosno prema distribuciji. Također, jedna od četiri distribucijske pumpe je stand-by pumpa, također označena plavom bojom. Na slici se također nalaze HMI komande za pokretanje „start“ i za zaustavljanje „stop“ hidrauličke jedinice.

## 6. ZAKLJUČAK

Sekvencijalno upravljanje je jedno od najčešćih načina upravljanja procesa. Sekvencijalnim upravljanjem izvršnim članovima u procesu zadajemo određeni niz naredbi npr. ( uključi/isključi i sl. ), pri čemu svaki proces traje unaprijed određeni vremenski interval. Sekvencijalno upravljanje procesima se uglavnom vrši pomoću PLC-ova. PLC-ovi su snažna i modularna računala namijenjena za rad u industriji. Predstavljaju središnje računalo u industrijskoj automatizaciji. Potrebno je bilo izraditi programski blok za automatsko pokretanje hidrauličke jedinice. Programski blok pri pokretanju hidrauličke jedinice vodi računa o tome da se pumpe pokreću sekvencijalno, o ispunjenosti uvjeta za uključivanje svake pojedine pumpe te o njihovom uključivanju ovisno o broju radnih sati svake pumpe. U slučaju iznenadnog gašenja neke od pumpi programski blok mora pokrenuti stand-by pumpu. Prije svakog pokretanja hidrauličke jedinice programski blok uspoređuje radna vremena pumpi te onu pumpu koja ima najveći broj radnih sati postavlja za stand-by pumpu. Pogodnost ovog programskog bloka je ta što postizemo podjednak broj radnih sati svih pumpi, te u slučaju ispada jedne od pumpi proces neće stati već će se automatski pokrenuti stand-by pumpa. Također pri pokretanju hidrauličke jedinice u automatskom režimu rada, pumpe se pokreću slijedno jedna za drugom. S tim onemogućujemo preopterećenje mreže. Vizualizacija programskog bloka je razvijena u programskom alatu CoDeSys.

## LITERATURA

- [1] SIEMENS, SIMATIC Working with STEP 7 Getting Started
- [2] Smart Software Solutions, User Manual for PLC Programming with CoDeSys 2.3
- [3] H. Berger, SIMATIC automatizacijski sustavi, SIEMENS, Graphis d.o.o., Zagreb, 2013., ISBN: 978-953-279-031-3
- [4] G. Malčić, Programirljivi logički kontroleri, Tehničko veleučilište u Zagrebu, Elektrotehnički odjel, Zagreb, 2007.
- [5] Benjamin Žagar, UPRAVLJANJE POZICIJOM U OTVORENOJ PETLJI POMOĆU DIGITALNIH DAVAČA POZICIJE TE DIGITALNIH I ANALOGNIH AKTUATORA, Diplomski rad, Osijek 2016.
- [6] Adrian Čičić, LINEARIZACIJA PROPORCIONALNOG VENTILA, Završni rad, Osijek 2017.
- [7] Tomislav Zubak,, MAKETA HIDRAULIČKE JEDINICE, Diplomski rad, Osijek 2017.

## SAŽETAK

U ovom radu je izrađen programski blok za automatsko pokretanje hidrauličke jedinice. Sustav se sastoji od n recirkulacijskih pumpi te m distribucijskih pumpi. Programski blok vodi računa o ispunjenosti uvjeta za uključivanje svake pojedine pumpe, te o njihovom uključivanju ovisno o broju radnih sati svake pojedine pumpe. Također po istom principu vodi računa o uključivanju rezervne pumpe u slučaju iznenadnog gašenja neke od glavnih pumpi. U programu CODESYS izrađena je vizualizacija uključivanja hidrauličke jedinice.

Ključne riječi: STEP7, CODESYS, ladder logic, automatizacija, hidraulička jedinica

## **ABSTRACT**

In this paper, a program block for automatic startup of a hydraulic unit was developed. The system consists of  $n$  recirculation pumps and  $m$  distribution pumps. The program block takes into account the fulfillment of terms for the start up of each pump, and their start up depending on the number of working hours of each pump. Based on the same principle, it also takes into account of the start up of a spare pump in the case of a sudden shutdown of some of the main pumps. The visualisation of start up of the hydraulic unit was made in program CODESYS.

Keywords: STEP7, CODESYS, ladder logic, automation, hydraulic unit.



## **ŽIVOTOPIS**

Luka Strinić rođen je 4. 4. 1996. godine u Vinkovcima. 2015. godine završio srednju tehničku školu „Ruđera Boškovića“ u Vinkovcima, smjer elektrotehničar, te godine 2015. upisao preddiplomski stručni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku smjer automatika. Stručnu praksu odradio u tvrtki Danieli-System d.o.o.

Informatički je pismen, te posjeduje vještine rada u alatima poput AutoCAD, MATLAB, Siemens SIMATIC STEP7.

Posjeduje vozačku dozvolu B kategorije, te se služi Engleskim jezikom u govoru i pismu.