



**EduSplit** Obrtna tehnička škola  
Regionalni centar kompetentnosti Split

# PLC

Priručnik za izradu korisničkog programa za programabilne logičke kontrolere



Marko Vukšić



Marko Vukšić

## PLC

Priručnik za izradu korisničkog programa za programabilne logičke kontrolere



Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Sadržaj publikacije isključiva je odgovornost Obrtne tehničke škole Split





**EduSplit** Obrtna tehnička škola  
Regionalni centar kompetentnosti Split

Marko Vukšić

# PLC

Priručnik za izradu korisničkog programa za programabilne logičke kontrolere

Split, 2023.

Autor: dr.sc. Marko Vukšić, prof. s. s. u t. i.  
Urednik: Igor Koletić, mag. ing. mech. (za ALGEBRU d. o. o.)  
Naslov: **PLC**  
Fotografija naslovnice: Freepik  
Recenzent: Ivan Jajčević, dipl. ing.  
Lektorica: Anamarija Knezović Kaurić, prof.  
Grafičko oblikovanje: ALGEBRA d.o.o.  
Ilustracije: izrada autora u više programa  
Nakladnik: Obrtna tehnička škola Split  
Odgovorna osoba: ravnatelj Milivoj Kalebić  
Za nakladnika: ALGEBRA d.o.o.

Više informacija:

Obrtna tehnička škola Split  
Plančićeva 21  
21000 Split  
e-pošta: ured@ss-obrtna-tehnicka-st.skole.hr  
mrežna adresa: edusplit.eu

ISBN: 978-953-50488-3-1

---

**Regionalni centar kompetentnosti Obrtne tehničke škole**

**Split, 2023.**

Obrtnička tehnička škola, Plančićeva 21, 21000 Split, OIB: 43651407703, nositelj je isključivog prava iskorištavanja ovog autorskog djela, prostorno, vremenski i sadržajno neograničeno, a koje pravo obuhvaća imovinska prava autora i to osobito, ali ne isključivo, pravo reproduciranja (pravo umnožavanja), pravo distribuiranja (pravo stavljanja u promet), pravo priopćavanja autorskog djela javnosti te pravo prerade. Pojedina imovinska autorska prava treća osoba može steći isključivo na temelju pisane suglasnosti Obrtničke tehničke škole.

# Sadržaj

<b>1. Uvod</b>	13
1.1. Ciklus obrade programa	13
1.2. Modul digitalnih ulaza	15
1.3. Modul digitalnih izlaza	16
1.4. Strujni izvor i strujni ponor	17
1.5. PNP i NPN senzor	19
1.6. Modul analognih ulaza i modul analognih izlaza	20
1.7. Mrežni protokoli	20
1.8. Simatic PLC S7-1200	21
1.8.1. Programabilni kontroler S7-1200	24
1.8.2. Pregled PLC modula	25
1.9. Programski jezici	26
1.10. Izvršenje korisničkog programa	28
<b>2. Varijable i tipovi podataka</b>	30
2.1. Operandi i tagovi	30
2.1.1. Pojmovi	30
2.1.2. Periferni ulazi i izlazi, procesni ulazi i izlazi	31
2.1.3. Memorijski bitovi	32
2.1.4. Podatkovni blokovi	33
2.2. Adresiranje	33
2.2.1. Putanja signala	33
2.2.2. Apsolutno adresiranje operanda	34
2.2.3. Simbolično adresiranje	36
2.2.4. Adresiranje konstanti	36
2.2.5. Indirektno adresiranje	36
2.3. Opće informacije o tipovima podataka	37
2.4. Elementarni tipovi podataka	37
2.4.1. BOOL, BYTE, WORD and DWORD	37
2.4.2. BCD-kodirani brojevi	38
2.4.3. USINT, UINT i UDINT, te SINT, INT and DINT	38
2.4.4. REAL i LREAL	39
2.4.5. CHAR	40
2.5. Sistemski podatci fiksne strukture	40
2.5.1. IEC tajmeri	40
2.5.2. IEC brojači	43
<b>3. Ljestvičasti dijagrami - programiranje</b>	46
3.1. Uvod	46

3.1.1. Osnova programiranja ljestvičastim dijagramima .....	47
3.1.2. Programske naredbe LAD .....	48
3.2. Programiranje relejnim kontaktima .....	48
3.2.1. NO i NC kontakt .....	49
3.2.3. Paralelni spoj .....	51
3.2.4. Serijsko-paralelni spoj .....	51
3.2.5. Kombinirani serijsko-paralelni spoj .....	52
3.2.6. T-grananje .....	53
3.2.7. Negacija rezultata logičke operacije .....	54
3.3. Programiranje svitcima .....	54
3.3.1. Standardni i negativni svitak .....	55
3.3.2. Set i reset svitak .....	55
3.3.4. IEC tajmer .....	56
3.4. Funkcije – usporedba, premještanje i konverzija .....	57
3.5. Aritmetičke funkcije za numeričke vrijednosti .....	58
3.6. Funkcije kontrole toka .....	60
3.6.1. Funkcija Jump .....	60
3.6.2. Poziv funkcijskog bloka .....	60
3.6.3. Poziv funkcije .....	62
<b>4. Zadaci za vježbu .....</b>	<b>64</b>
4.1. Zadatak 1: Temeljne funkcije ljestvičastog dijagrama .....	64
4.1.1. Zadatak .....	64
4.1.1. Rješenje .....	65
4.2. Zadatak 2: Trčeće svjetlo .....	71
4.2.1. Zadatak .....	71
4.2.2. Rješenje .....	72
4.3. Zadatak 3: Semafor .....	78
4.3.1. Zadatak .....	78
4.3.2. Rješenje .....	82
4.4. Zadatak 4: Mjerenje protoka .....	86
4.4.1. Zadatak .....	86
4.4.2. Rješenje .....	87
Popis slika .....	91
Popis tablica .....	93
Popis korištenih kratica .....	94
Rječnik pojmova .....	96
Popis literature .....	101





## 1. Uvod

Programabilni logički kontroler (engl. **Programable Logic Controller**), skraćeno PLC, digitalno je računalo koje za osnovnu funkciju ima upravljanje i nadzor strojeva i industrijskih procesa.

Vizualizacija nadzora provodi se preko **HMI** panela (engl. **Human Machine Interface**).

PLC je nastao kao zamjena za relejne sustave na kojima se osnivala tadašnja logika upravljanja. Najveća mana tih sustava je složenost izvedbe, otežana dijagnostika u slučaju kvara, kao i dugotrajno i skupo osposobljavanje sustava nakon kvara. Ukazala se potreba da se elektromehanički dijelovi, kao logički elementi, zamijene poluvodičkim digitalnim računalom sposobnim odraditi funkcije povezanih releja za obavljanje logičkih zadataka.

PLC treba ispunjavati sljedeće uvjete:

- godinama pouzdano raditi
- biti konfigurabilan
- biti otporan na električne smetnje
- biti otporan na vibracije
- raditi unutar zadanog temperaturnog opsega.



Slika 1.1. Modularni PLC različitih proizvođača



Slika 1.2. Kompaktni PLC tvrtke SIEMENS

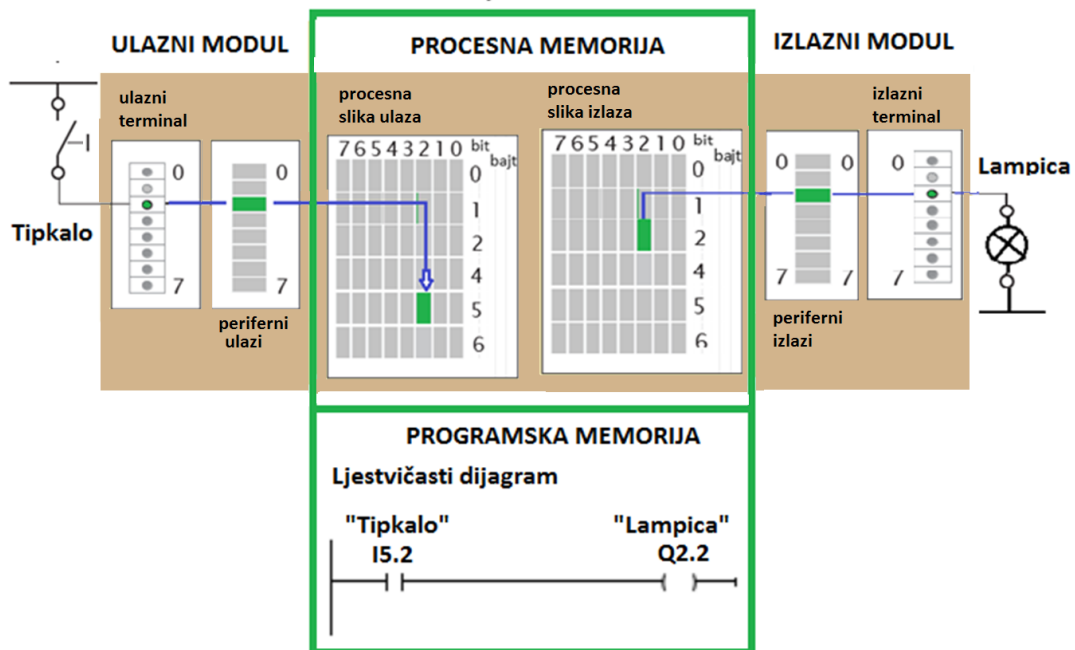
Svi PLC-ovi imaju iste osnovne komponente, a to su (slika 1.3):

- procesor ili središnja procesorska jedinica CPU (engl. **C**entral **P**rocessing **U**nit)
- ulazni i izlazni moduli (I/O)
- napajanje
- komunikacijski moduli
- periferni uređaji.

## CPU

CPU je „mozak“ PLC-a, a sastoji se od memorije i integriranih sklopova koji neprestano pokreću memorijske rutine te:

- izvršavaju upravljačke naredbe pohranjene u programskoj memoriji - upravljačka logika
- nadziru proces i komuniciraju s drugim uređajima.



Slika 1.3. Načelna blok-shema PLC-a

Memorija osigurava pohranu operativnog sustava i firmvera procesora i modula te programa i podataka. Programska memorija preuzima upravljački program pomoću uređaja za programiranje ili računala s odgovarajućim softverom za programiranje.

Podatkovna memorija pohranjuje procesne slike ulaza i izlaza kao skup analognih i digitalnih vrijednosti.

### Ulazno-izlazni moduli (I/O)

I/O moduli mogu biti pojedinačne I/O kartice ili složene decentralizirane I/O periferije.

Ulazni moduli prikupljaju signale s ulaznih uređaja, primjerice, senzora, prekidača i sl.

Izlazni moduli prosleđuju komande na izlazne uređaje, primjerice, lampice, kontaktore, sklopnike, motore, elektromagnetske ventile.

### Napajanje

Svaki PLC ima pripadajuće napajanje potrebno za njegov rad i rad njegovih perifernih modula.

### Komunikacijski moduli

Putem različitih vrsta sabirnica PLC komunicira s različitim „inteligentnim“ uređajima, na primjer, frekvencijskim pretvaračima, kamerama, drugim PLC-ovima, sensorima.

### Periferni uređaji

Periferni uređaji spojeni su na PLC putem I/O modula odgovarajućeg naponskog nivoa ili putem komunikacijskog modula s odgovarajućom sabirnicom. Primjerice, inkrementalni enkoder spojen preko odgovarajućeg brzog brojačkog modula, mjerna ćelija spojena PRO-FINET mrežom.

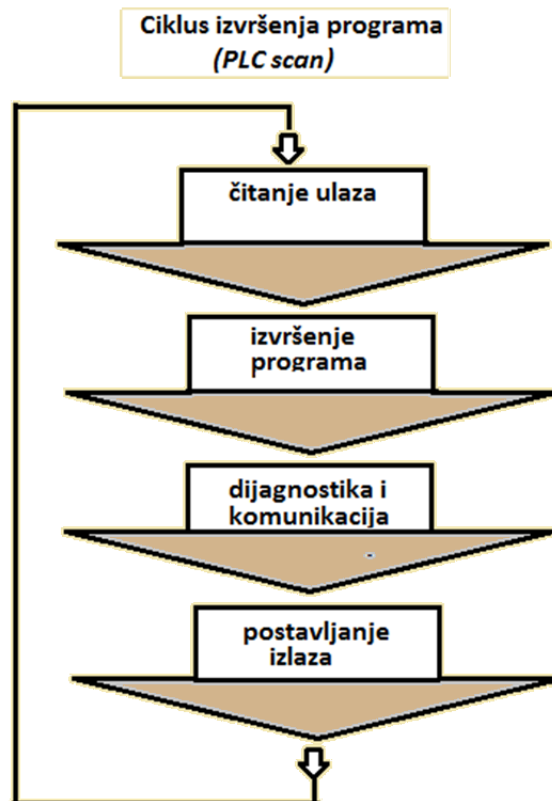
## 1.1. Ciklus obrade programa

Princip rada PLC uređaja je sljedeći: kontinuirano prateći ulazne informacije iz procesa, a poštujući zadanu logiku rada, program postavlja izlaze PLC-a u odgovarajuća stanja te, na taj način, informacije prosleđuje natrag u proces.

Jedan ciklus obrade programa nazivamo **skan** ciklus (slika 1.4). Skan ciklus ovisi o veličini korisničkog programa i njegovoj organizaciji i u većini slučajeva je kraći od nekoliko milisekundi. Sastoji se od četiri glavna koraka:

- čitanje ulaza - status svih ulaznih modula upisuje se u memorijsko područje koje se naziva procesna slika ulaza

- izvršenje programa - podatci iz procesne slike ulaza obrađuju se logikom koja je programirana u korisničkom programu, nakon čega se ažurira procesna slika izlaza
- dijagnostika i komunikacija - obrada internih vrijednosti brojača i tajmera, obrada alarmnih stanja
- postavljanje izlaza - podatci iz procesne slike izlaza prosljeđuju se svim izlaznim modulima koji su dio sustava.



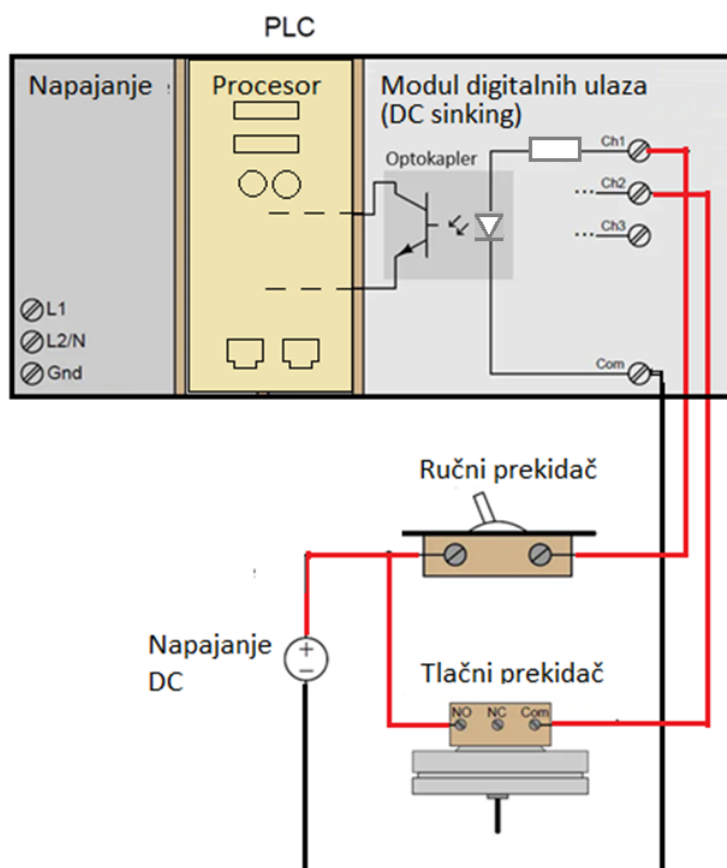
Slika 1.4. Jedan skan ciklus

Svi proizvođači PLC opreme i pripadajućih programskih alata pridržavaju se **IEC** (engl. *International Electrotechnical Commission*) standarda IEC 61131 te su znanja stečena radom na opremi jednog proizvođača lako prenosiva i na rad s PLC-om drugog proizvođača. Standardom su definirane glavne karakteristike PLC uređaja te povezane periferne opreme, način komunikacije između PLC-ova i druge opreme te pravila za korištene programske jezike. Dio III ovog standarda specificira sintaksu i semantiku unificiranog skupa programskih jezika za PLC-ove.

## 1.2. Modul digitalnih ulaza

Sklopke, tipkala, granične sklopke i induktivni senzori primjeri su diskretnih senzorskih uređaja. Diskretna podatkovna točka ona je točka koja ima samo dva stanja: uključeno i isključeno.

Interni shematski dijagram za modul digitalnih ulaza (slika 1.5) otkriva komponente tipične za jedan ulazni kanal na toj kartici. Moduli digitalnih ulaza obično imaju 4, 8, 16 ili 32 kanala.

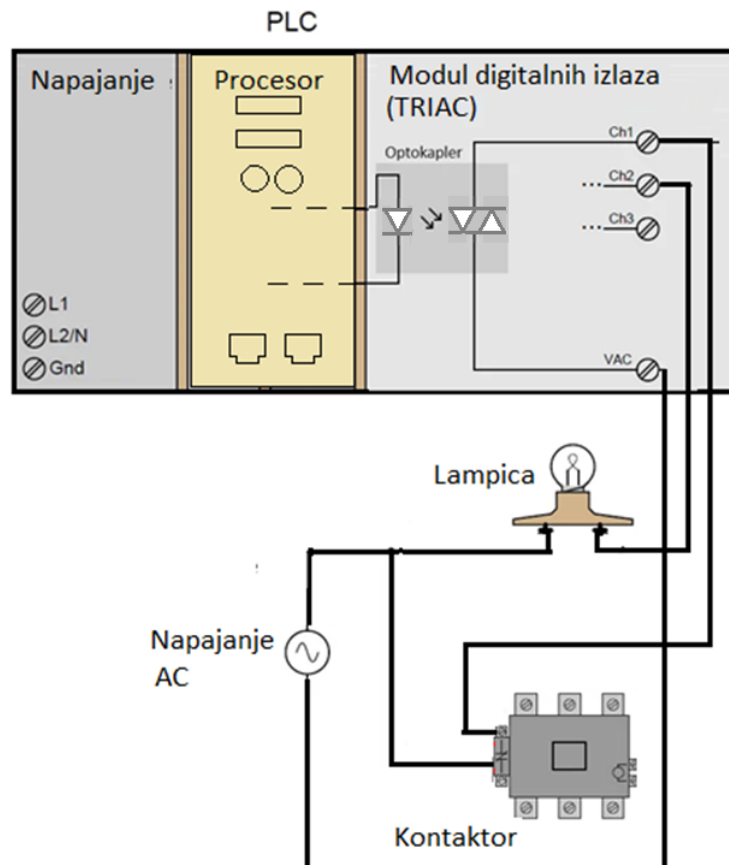


Slika 1.5. Modul digitalnih ulaza

Da bi PLC bio svjestan stanja diskretnog senzora, mora primiti signal od senzora kroz diskretni kanal ulaznog modula. Unutar svakog modula digitalnih ulaza obično se nalazi skup svijetlećih dioda (LED) koje će se uključiti kada se uključi odgovarajući senzorski uređaj. Svjetlo svake LED diode obasjava fotoosjetljivi uređaj kao što je fototranzistor unutar modula koji, zauzvrat, aktivira bit (jedan element digitalnih podataka) unutar memorije PLC-a. Ta optička sprega (engl. *optocoupler*) svaki ulazni kanal modula digitalnih ulaza PLC-a čini prilično robusnim, sposobnim izolirati osjetljivi računalni sklop PLC-a od prolaznih „šiljaka“ napona i drugih električnih fenomena koji mogu uzrokovati štetu na elektroničkom sklopu.

### 1.3. Modul digitalnih izlaza

Indikatorske lampice, solenoidni ventili i pokretači motora (sklopovi koji se sastoje od kontaktora i uređaja koji štite od preopterećenja) primjeri su diskretnih upravljačkih uređaja. Interni shematski dijagram za modul digitalnih izlaza (slika 1.6) otkriva komponente tipične za jedan izlazni kanal na toj kartici. Moduli digitalnih izlaza obično imaju 4, 8, 16 ili 32 kanala.



Slika 1.6. Modul digitalnih izlaza

Slično kao u slučaju diskretnih ulaza, PLC se povezuje s bilo kojim brojem različitih diskretnih završnih upravljačkih uređaja putem diskretnog izlaznog kanala. Svaki izlazni kanal ima vlastitu optičku spregu i upravljan je vlastitim jedinstvenim bitom memorijskog registra unutar memorije PLC-a.

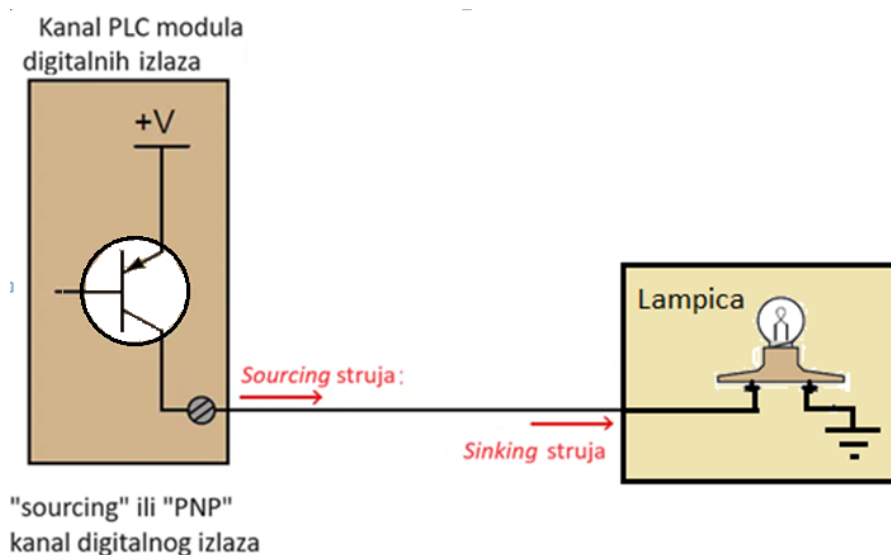
Moduli digitalnih izlaza obično upotrebljavaju isti oblik optoizolacije kako bi omogućili PLC-ovim računalnim sklopovima da šalju električnu energiju potrošačima: interni PLC sklop pokreće LED diodu koja zatim aktivira neki oblik fotoosjetljivog sklopnog uređaja.

Alternativno, mali elektromehanički releji mogu se koristiti umjesto optoizolacijskih sklopnih poluvodičkih elemenata kao što su tranzistori (DC) ili TRIAC (AC).

## 1.4. Strujni izvor i strujni ponor

U radu s modulima važno je razlikovati digitalne ulaze i uređaje sa strujnim izvorom i strujnim poniranjem.

Npr. na sljedećoj slici (slika 1.7) prikazuje se izlazni kanal PLC-a koji dovodi struju do indikatorske lampice, a ona struju propušta prema masi.



Slika 1.7. Sinking i sourcing

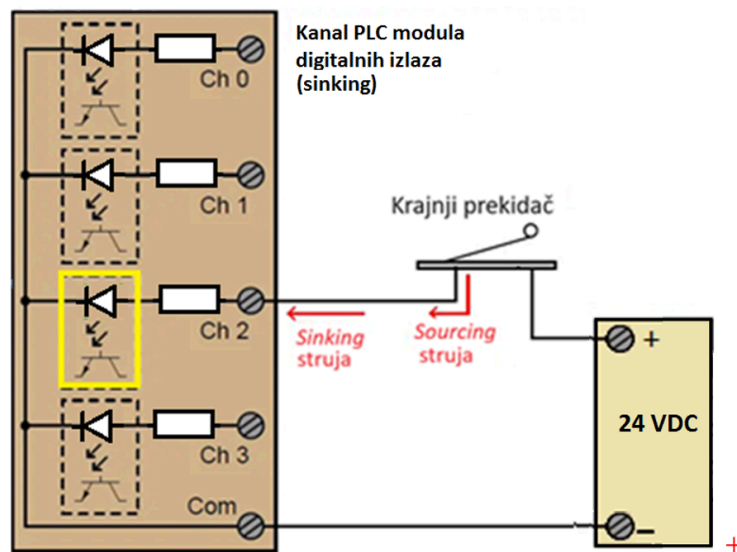
Pojmovi „izvor“ (engl. *sourcing*) i „poniranje“ (engl. *sinking*) odnose se na smjer struje, kao što je i označeno konvencionalnim zapisom protoka ( $\rightarrow$ ) u ili iz upravljačkog vodiča uređaja. Upravljački vodič povezuje kanal I/O kartice s uređajem polja, za razliku od vodiča koji su izravno povezani s pozitivnim ili negativnim vodom izvora napona.

Uređaj, koji šalje struju iz svog upravljačkog priključka na neki drugi uređaj, naziva se strujni izvor. Uređaj, koji prihvaća struju u svoj upravljački priključak, zovemo strujnim ponorom.

Ovi pojmovi imaju smisla ako se električna struja promatra iz perspektive konvencionalnog toka gdje je pozitivni terminal istosmjernog napajanja zamišljen kao izvor struje ( $[\rightarrow]$ ), a struja pronalazi svoj put do negativnog pola terminala istosmjernog napajanja ( $\rightarrow[ ]$ ).

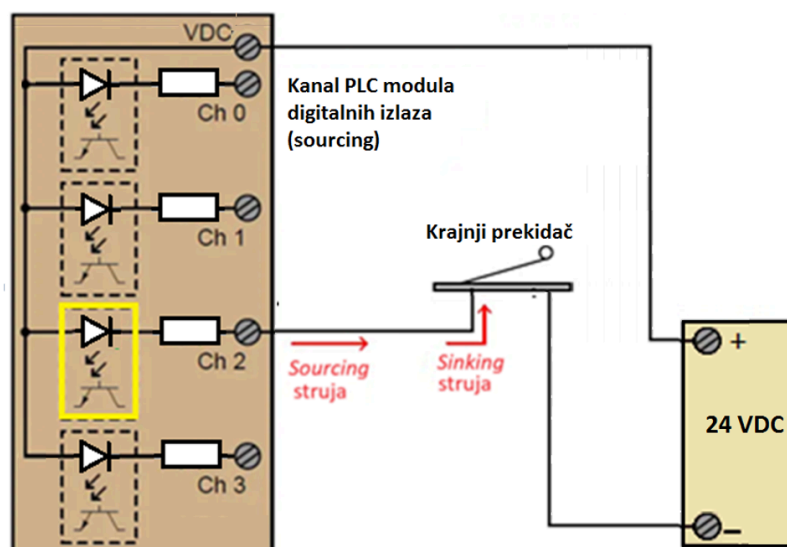
Primjerice, prikazuje se mehanička granična sklopka koja se spaja na PLC ulaz kao strujni ponor (slika 1.8), i na PLC ulaz kao strujni izvor (slika 1.9).

Važno je obratiti pažnju na pravilno povezivanje kanala na modulu digitalnih ulaza, ovisno o vrsti PLC ulaza - strujni izvor ili strujni ponor. Razlike u polaritetu i označavanju terminala naznačene su na slikama.



Slika 1.8. PLC kanal kao strujni ponor

Kod PLC kanala koji je strujni ponor, terminal ulaznog kanala („Ch2“) pozitivan je dok je zajednički terminal („Com“) negativan.

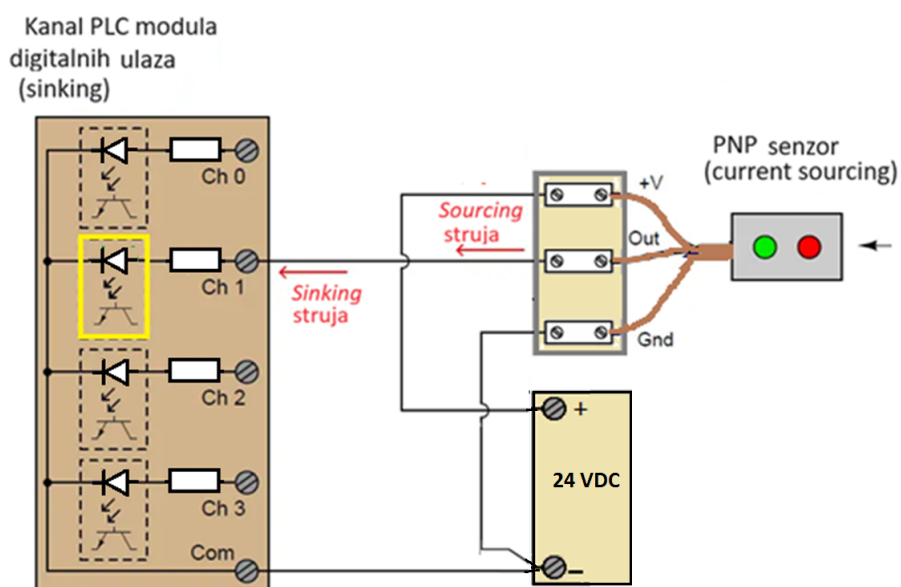


Slika 1.9. PLC kanal kao strujni izvor

Kod PLC kanala koji je strujni izvor, priključak ulaznog kanala („Ch2“) negativan je dok je zajednički priključak („VDC“) pozitivan.

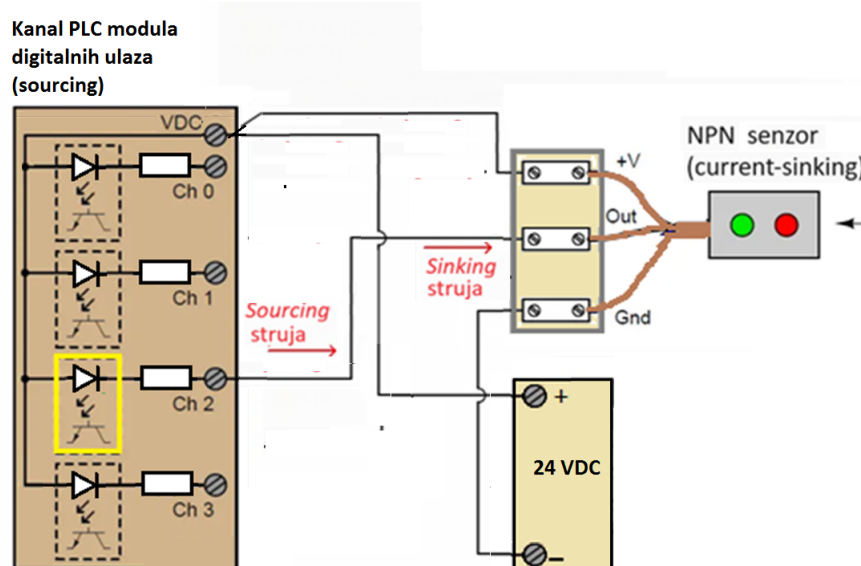
## 1.5. PNP i NPN senzor

Neki diskretni senzorski uređaji osjetljivi su na polaritet, poput elektroničkih senzora koji sadrže tranzistorske izlaze te se, kao takvi, mogu spajati samo s kompatibilnim PLC ulaznim kanalom. Ako je senzor tipa PNP, (strujni izvor) spaja se na PLC kanal koji je strujni ponor (slika 1.10).



Slika 1.10. PNP senzor

Ako je senzor tipa NPN (strujni ponor), spaja se na PLC kanal koji je strujni izvor (slika 1.11).



Slika 1.11. NPN senzor

## 1.6. Modul analognih ulaza i modul analognih izlaza

Analogni signali kontinuirani su i promjenjivi signali.

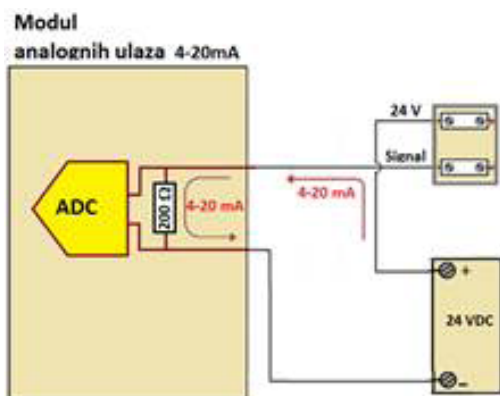
PLC moduli analognih ulaza obrađuju analogne signale. Da bi se uspostavilo sučelje s analognim senzorom ili kontrolnim uređajem, potreban je neki „prijevod“ između analognog i digitalnog svijeta. Unutar svakog analognog ulaznog modula nalazi se ADC ili analogno-digitalni pretvarač, sklop dizajniran za pretvaranje analognog električnog signala u višebitnu binarnu riječ.

PLC moduli analognih izlaza postavljaju analogne signale na svoje izlaze. Svaki analogni izlazni modul sadrži DAC ili digitalno-analogni pretvarač, krug za pretvaranje digitalnih naredbenih riječi PLC-a u analogne električne veličine.

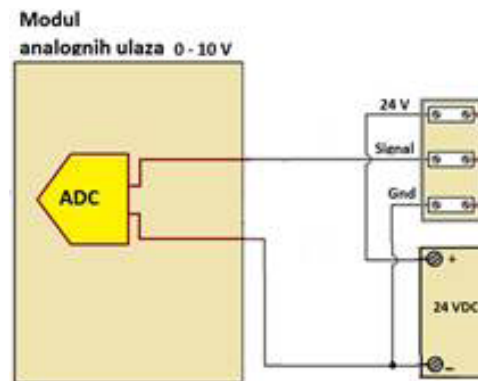
Moduli analognih ulaza i moduli analognih izlaza obično imaju 2, 4, ili 8 kanala.

Postoji više tipova modula analognih ulaza i modula analognih izlaza. Prilagođeni su radu za različite vrste analognih signala, uključujući:

- napon (0 - 10 volti,  $\pm 10$  volti,  $\pm 2.5$  volta,  $\pm 5$  volti)
- struju (0 do 20 mA, 4 do 20 mA)
- termopar (tip N NiCr-NiSi, tip E NiCr-CuNi, tip K NiCr-Ni, ...)
- RTD (Pt100, Ni100, Cu10, ...).



Slika 1.12. Modul analognih ulaza 4-20mA



Slika 1.13. Modul analognih ulaza 0-10V

## 1.7. Mrežni protokoli

Postoje različiti mrežni komunikacijski protokoli kojima PLC-ovi komuniciraju - od PLC-a do PLC-a i između PLC-a i uređaja u procesnom polju.

Najčešće korišteni protokoli, a ujedno i najpoznatiji su:

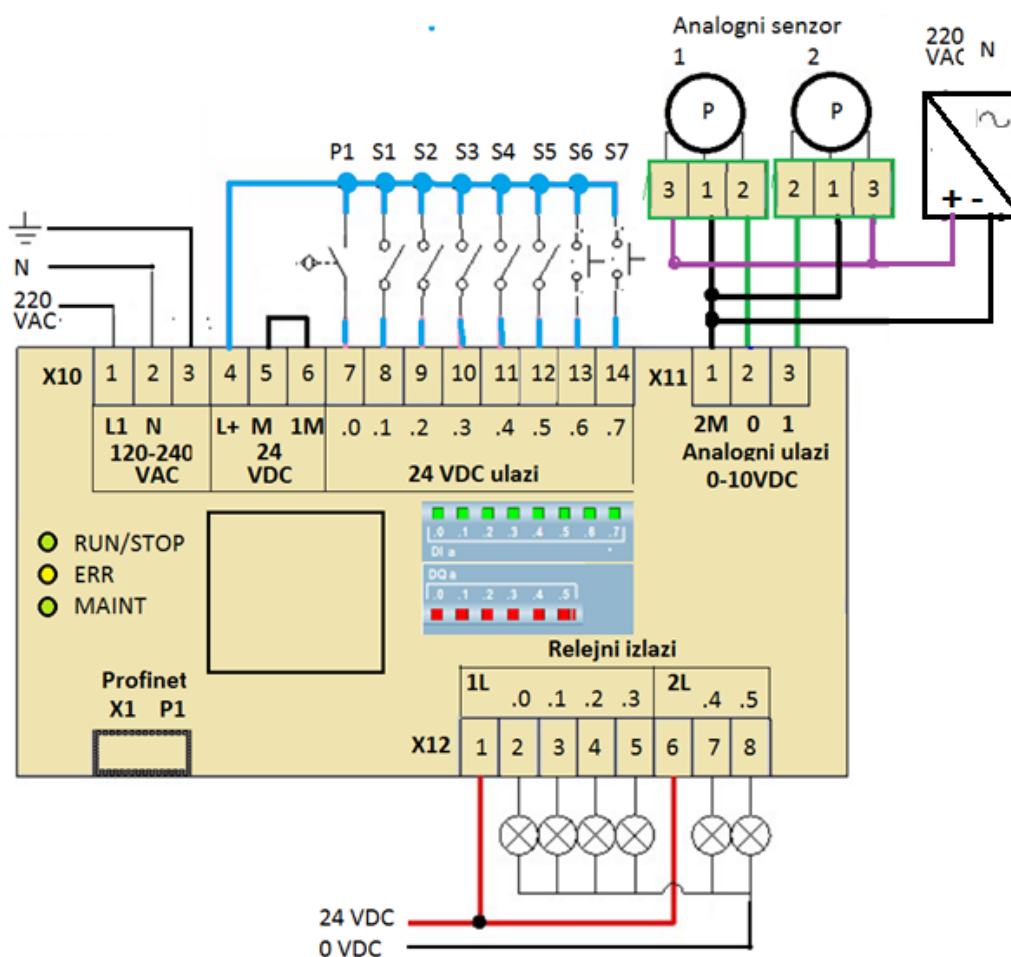
- Modbus protokol, izvorno za Modicon marku PLC-a
- PROFIBUS, koji je izvorno razvila tvrtka Siemens
- Profinet.

## 1.8. Simatic PLC S7-1200

Ovaj priručnik temelji se na IEC standardu 61131 koji definira električne i programske standarde PLC opreme. Sve navedeno primjenjivo je i za PLC proizvodnje Siemens (slika 1.14), ali nije ograničeno isključivo na opremu tog proizvođača.

Primjeri programskog koda i sustava upravljanja koji se ovdje obrađuju, realizirani su i testirani na opremi:

- PLC kontroleru serije Simatic S7-1200
- HMI osnovnom panelu
- STEP7 programskom alatu
- SIMATIC NET mreži.



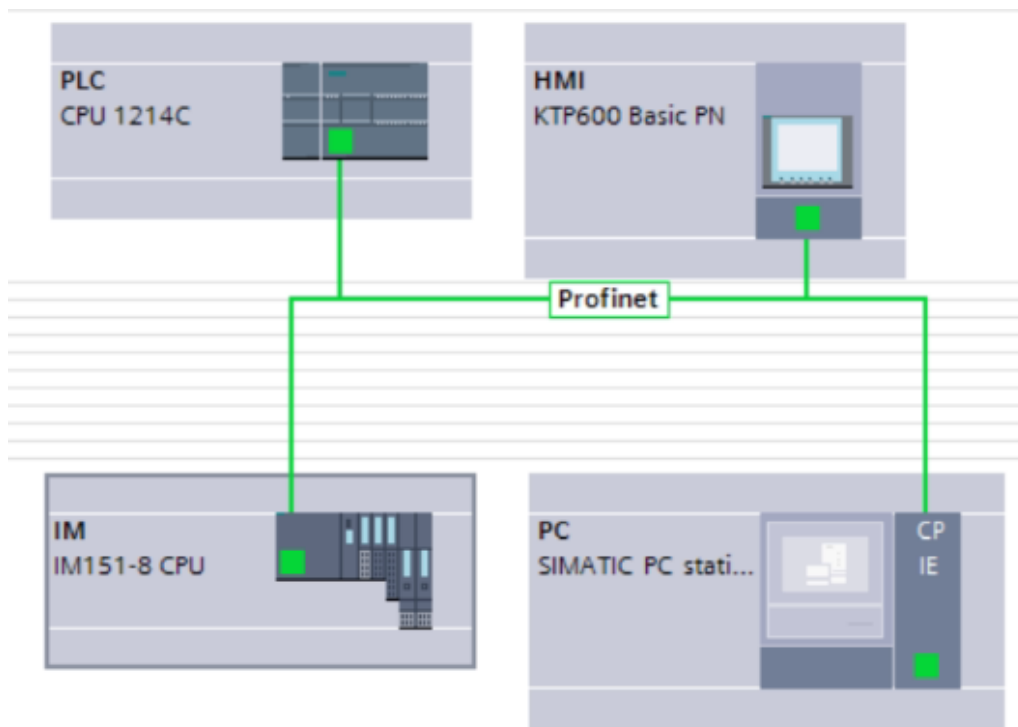
Slika 1.14. PLC S-1200 i primjer povezivanja senzora i aktuatora

**PLC S7-1200** „mozak“ je sistema upravljanja. Postoji više različitih verzija ovog uređaja, a razlikuju se po veličini memorije, broju digitalnih i analognih ulaza i izlaza te broju mrežnih priključaka.

**HMI panel** namijenjen je kontroli i nadzoru. Dije se ovisno o:

- mogućnostima - na Basic i Comfort
- veličini - na 4", 7", 9", 12" za Basic te 4", 7", 9", 12", 15", 19" i 22" za Comfort
- načinu ugradnje - na fiksni i mobilni
- izvedbi - na Touch screen ili Key panel.

**STEP 7 program** koristi se za konfiguraciju, programiranje i parametiranje sustava. U okruženju nazvanom **TIA portal** (*Totally Integrated Automation*) daje se alat kojim se sinkronizirano programira, kako PLC tako i HMI panel, te omogućava komunikacija s drugim dijelovima sustava. Moguća je i simulacija sustava u svrhu testiranja programa i prije njegove implementacije.



Slika 1.15. Komponente SIMATIC S7-1200 sustava

**Simatic NET** mreža (slika 1.15) povezuje sve dijelove sustava uz kontrolu sigurnosti mrežnog prometa. Dodatnim komunikacijskim modulima mreža se konfigurira na neki od standarda:

- PROFINET
- PROFIBUS DP
- AS-Interface
- MODBUS
- *point-to-point*.

## PROFINET

Koristi se za komunikaciju s drugim uređajima (npr. HMI panelima, sensorima i sl.) preko etherneteta. PROFINET je otvoreno industrijsko ethernet rješenje temeljeno na međunarodnim standardima. Ovisno o verziji CPU, PLC S7-1200 može komunicirati preko mreže s maksimalno 16, odnosno 12 PROFINET uređaja. S7-1200 na sebi ima ugrađen PROFINET priključak. Predstavljen je početkom 2000-ih i najbolje je prihvaćeno industrijsko ethernet rješenje. Budući da je PROFINET otvoreni standard, stotine proizvođača razvilo je PROFINET proizvode, poput PLC-ova, PAC-ova, elektromotornih pogona, robota, proxyja, IO-a i dijagnostičkih alata. PROFINET definira cikličku i acikličku komunikaciju između uređaja, uključujući dijagnostiku, funkcionalnu sigurnost, alarme i dodatne informacije.

Za povezivanje svih tih komponenti PROFINET koristi standardni ethernet za svoj komunikacijski medij. Ethernet kabeli povezuju PROFINET komponente unutar mreže, dopuštajući drugim ethernet protokolima koegzistiranje unutar iste infrastrukture.

## PROFIBUS DP

PROFIBUS DP je vrsta protokola i sabirnice koji omogućava komunikaciju između glavnog upravljačkog uređaja i određenog broja podređenih uređaja. Fizički sloj temelji se na standardu RS-485, istom fizičkom sloju kao i Modbus, ali ima specijalizirani protokol koji se naziva *token ring* koji omogućava priključenje više glavnih uređaja na istu komunikacijsku mrežu, čime se razlikuje od ostalih sabirničkih mreža. Token se prenosi između nadređenih uređaja, a koji god nadređeni uređaj drži token, ima ekskluzivnu mogućnost komuniciranja s podređenim uređajima na mreži.

PLC se dograđuje komunikacijskim procesorom ovisno o tome želi li se da radi kao PROFIBUS DP Master (CM1243-5) ili PROFIBUS DP slave (CM1242-5). Moguće je istovremeno spojiti nekoliko komunikacijskih modula i, na taj način, međusobno odvojiti PROFIBUS mreže po funkcionalnim cjelinama.

## ASI

ASI (engl. **A**ctuator **S**ensor **I**nterface) sustav je u kojem postoji jedan nadređeni uređaj. Jedan glavni uređaj (ASI Gateway) razmjenjuje podatke sa svim uređajima u mreži. Prijenosni medij je dvožilni polarizirani žuti ravni kabel koji nije oklopljen. Kabel se istovremeno koristi za prijenos signala i isporuku energije. Moguće je izravno napajanje komunikacijske elektronike i sudionika u mreži koji imaju ograničene zahtjeve za napajanje. Odvojena, crna ravna žica može se koristiti za napajanje uređaja koji zahtijevaju dodatnu energiju poput pokretača motora.

PLC se dograđuje komunikacijskim procesorom (CM1243-2).

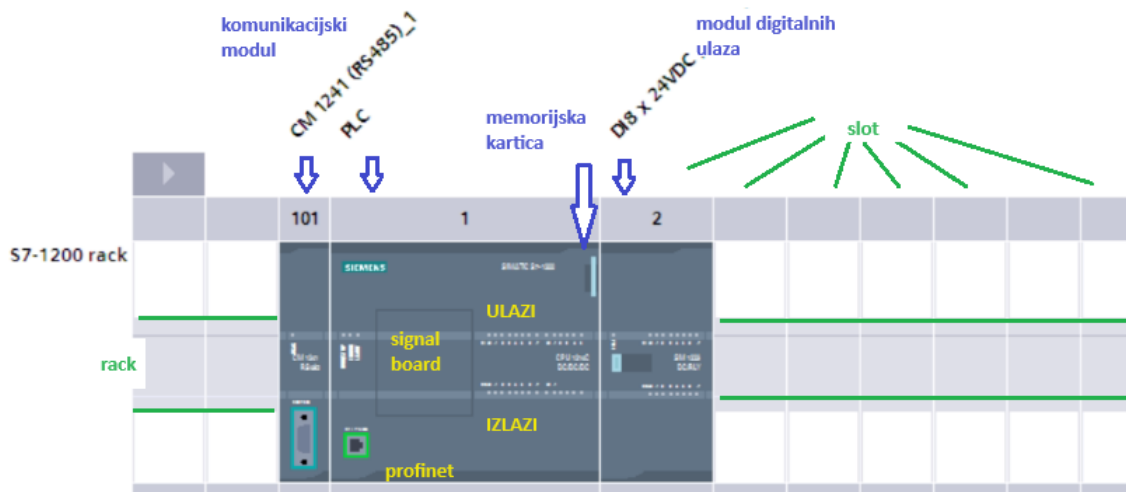
## MODBUS

MODBUS je vrsta protokola i sabirnice koji omogućava komunikaciju između dva uređaja od kojih je jedan nadređeni, a drugi podređeni. Fizički sloj temelji se na standardu RS-485. U odnosu nadređeni/podređeni, komunikacija se uvijek odvija u paru - jedan uređaj mora pokrenuti zahtjev i zatim čekati odgovor - a inicirajući uređaj (nadređeni) odgovoran je za pokretanje svake interakcije. Tipično, nadređeni uređaj je sučelje čovjeka i stroja (HMI) ili sustav nadzorne kontrole i prikupljanja podataka (SCADA), a podređeni je senzor, programibilni logički kontroler (PLC) ili frekvencijski pretvarač. U komunikaciji PLC - frekvencijski pretvarač, PLC uređaj je nadređeni, a frekvencijski pretvarač je podređeni uređaj. Sadržaj ovih zahtjeva i odgovora, te mrežni slojevi preko kojih se te poruke šalju, definirani su različitim slojevima protokola.

### 1.8.1. Programabilni kontroler S7-1200

PLC Simatic S7-1200 je modularne izvedbe (Slika 1.16) i sastoji se od:

- središnje procesorske jedinice **CPU** (engl. **C**entral **P**rocessing **U**nit) i sadrži operativni sistem i korisnički program
- signalnih modula, modula digitalnih i analognih ulaza i izlaza
- funkcijskih modula - izvršavaju vremenski kritične poslove nezavisno o CPU-u, primjerice, brojački modul
- komunikacijskog procesora **CP** (engl. **C**ommunication **P**rocessor), primjerice, PROFIBUS.



Slika 1.16. Opcije povezivanja na PLC S7-1200

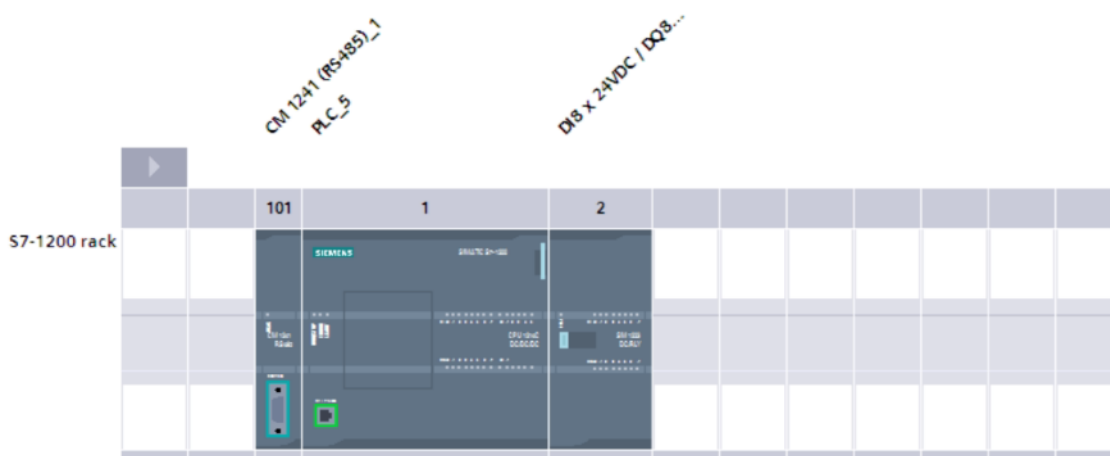
Memorijska područja kojima se koristi CPU:

- memorija za učitavanje (engl. *Load Memory*) - nalazi se na memorijskoj kartici **MMC** (engl. **M**icro **M**emory **C**ard) i upotrebljava se za smještaj upravljačkog programa. Može biti različitog kapaciteta

- radna memorija (engl. *Work Memory*) - integrirana je u CPU i koristi se za smještaj podataka potrebnih za izvršenje programa. Nema mogućnost dodatnog proširenja, određena je tipom CPU-a
- sistemska memorija (engl. *System Memory*) - integrirana je u CPU i koristi se za sliku ulaza (engl. *Input Image Memory*), za sliku izlaza (engl. *Output Image Memory*), za interne rezultate (engl. *Internal Memory* ili *Bit Memory*) za ulaze i izlaze, za međurezultate, za čuvanje privremenih podataka te za tajmere i brojače
- retentivna memorija (engl. *Retentive Memory*) - integrirana je u CPU i koristi se za čuvanje određenih podataka pri nestanku napajanja i ponovno pokretanje CPU-a.

### 1.8.2. Pregled PLC modula

Na samom PLC-u S7-1200 nalazi se određen broj digitalnih i analognih ulaza te digitalnih izlaza za konekciju s procesom (Slika 1.17). Njihov broj ovisan je o verziji CPU-a. Ti ulazi dizajnirani su za funkciju HSC (engl. **H**igh **S**peed **C**ounter) brzih brojačkih ulaza, a izlazi za funkciju pulsnih izlaza (engl. *Pulse*) primjerenu, npr., za upravljanje koračnim motorima.



Slika 1.17. Pregled PLC modula

Signalna kartica **SB** (engl. **S**ignal **B**oard) opcija je za proširenje, primjerice, dodatnim digitalnim ulazima ili s dodatnim analognim izlazima.

Kartica za komunikaciju **CB** (engl. **C**ommunication **B**oards) služi za omogućavanje *point-to-point* konekcije u slučaju da je potrebna. Za, primjerice, RFID čitač.

Komunikacijski modul **CM** (engl. **C**ommunication **M**odules) koristi se za povezivanje s nekom drugom vrstom mreže, npr., sa PROFIBUSOM.

Signalni moduli **SM** (engl. **S**ignal **M**odules) mogu biti nadodani u količini koji dopušta određena verzija CPU-a. Ovdje se radi o modulima digitalnih i analognih ulaznih i izlaznih signala.

Na samom PLC-u ugrađen je i **PROFINET** priključak za konekciju na mrežu na koju se može spojiti, npr., OP Panel, drugi PLC. PROFINET je standard koji se zasniva na TCP/IP protokolu.

## 1.9. Programski jezici

Postoji nekoliko programskih jezika koji se koriste pri programiranju PLC-a. U upotrebi su tri grafička programska jezika i dva tekstualna programska jezika.

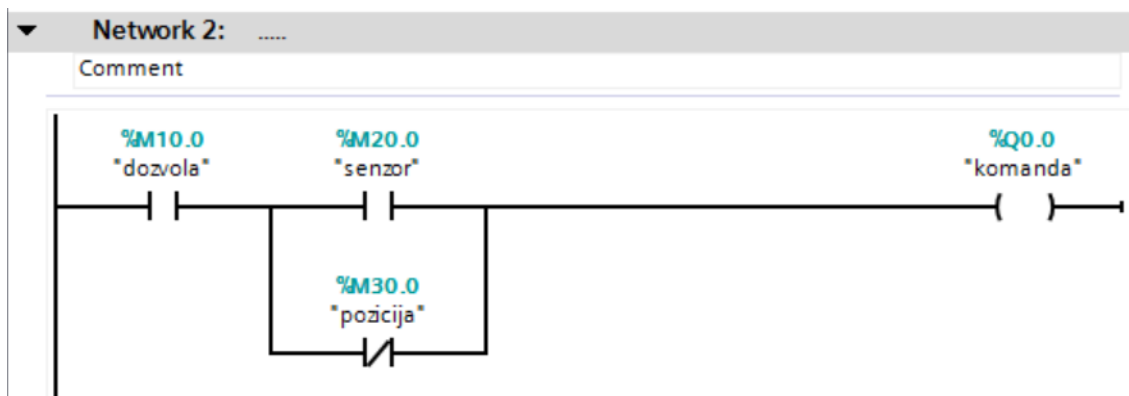
U programskom alatu TIA portal omogućeno je korištenje sljedećim programskim jezicima:

- ljestvičasti dijagram LAD (engl. **Ladder Logic**)
- dijagram funkcijskih blokova FBD (engl. **Function Block Diagram**)
- strukturirani upravljački jezik SCL (engl. **Structured Control Language**).

Korisnički program, u svakom svom dijelu, može se programirati u nekom od ovih jezika, kako korisniku za najbolje odgovara za određenu funkciju.

### LAD

**Ljestvičasti dijagram** grafički je programski jezik temeljen na relejnoj logici (slika 1.18). Ostvaruju se logičke funkcije na binarnim stanjima signala, stanje 0 ili 1, prikazanih kao serijski ili paralelno povezane kontakte odnosno kao „I” ili „LI” povezane kontakte. Elementi kruga (primjerice, normalno otvoren kontakt ili normalno zatvoren kontakt) kombiniraju se u formi mreže (engl. *Network*). Programirana sekcija može se realizirati pomoću jednog ili više blokova.

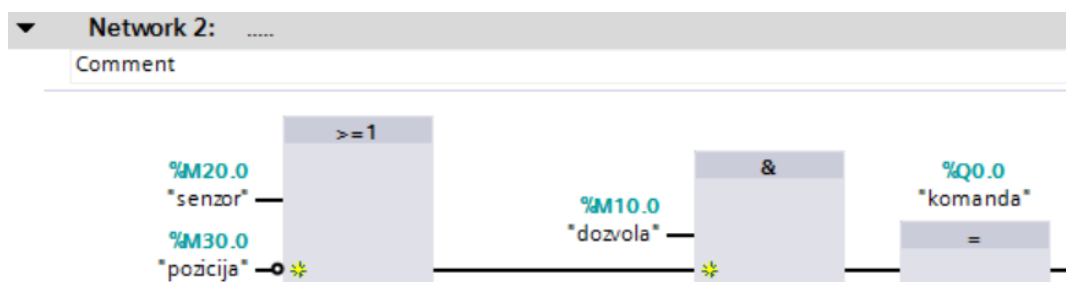


Slika 1.18. Ljestvičasti dijagram LAD

Složenije funkcije prikazane su simbolički, pomoću „crnih kutija”, npr. matematičke funkcije.

## FBD

**Dijagram funkcijskih blokova** (slika 1.19) grafički je programski jezik temeljen na prikazu sustava elektroničkih sklopova u obliku grafičkog bloka. Složenije funkcije također se prikazuju na isti način.



Slika 1.19. Dijagram funkcijskih blokova FBD

## SCL

**Strukturirani upravljački jezik** pogodan je za složene algoritme (slika 1.20). Sličan je programskom jeziku PASCAL.

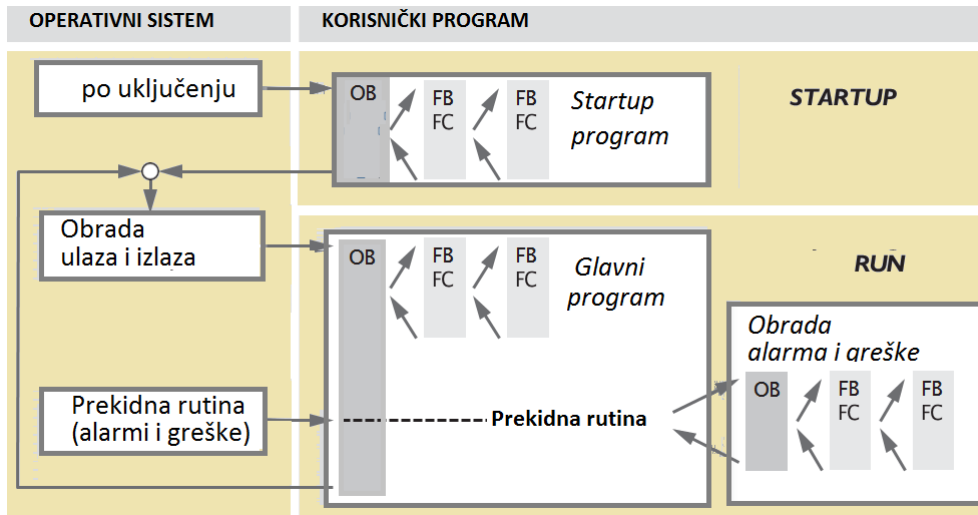
```
1 IF _condition_ THEN
2     // Statement section IF
3     ;
4 END_IF;
5
6 WHILE _condition_ DO
7     // Statement section
8     ;
9 END_WHILE;
10
```

Slika 1.20. Strukturirani upravljački jezik SCL

Pogodan je za jednostavne programske petlje i uvjetna grananja, za složene algoritme kao i za rad s većim blokovima podataka.

## 1.10. Izvršenje korisničkog programa

**START-UP mode:** po uključenju napajanja pokreće se „Program za pokretanje“, CPU provjerava komponente sustava i njihovu kompatibilnost kao i parametre sustava.

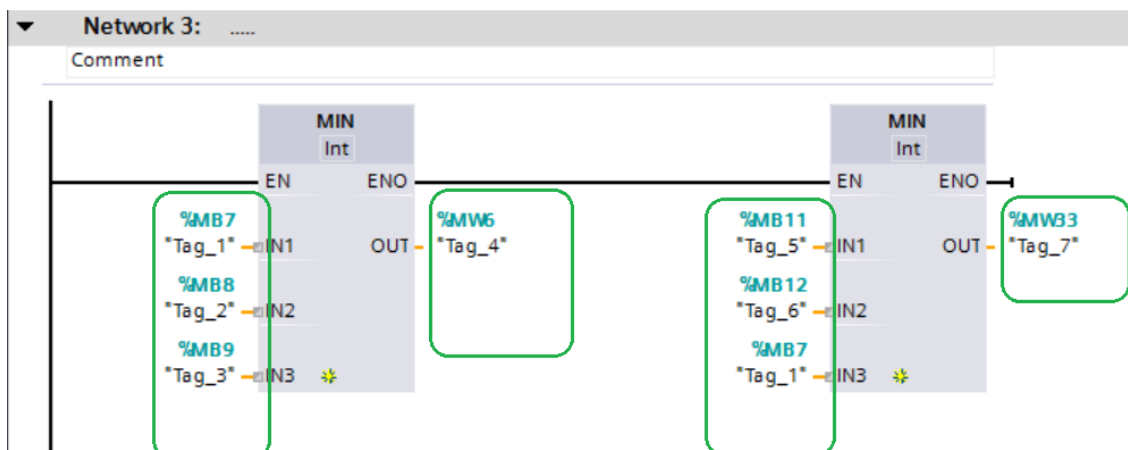


Slika 1.21. Izvršenje korisničkog programa

**RUN mode:** nakon programa za pokretanje pokreće se korisnički program. Korisnički program podijeljen je u blokove (slika 1.21):

- organizacijski blok **OB** (engl. **Organization Block**) obavezan je i predstavlja vezu između operativnog sistema i korisničkog programa
- funkcijski blok **FB** (engl. **Function Block**) - koristi se za izradu programa, ima memoriju u koju sprema lokalne podatke. Varijable se nalaze u podatkovnim blokovima na koje se referira određeni funkcijski blok (engl. *instance data block*)
- funkcija FC (engl. **Function**) - koristi se za izradu programa, nema dodijeljen trajni memorijski prostor za varijable, pristupa globalnim podatkovnim blokovima (engl. *global data block*),
- podatkovni blokovi DB (engl. **Data Block**) - dijelovi memorije u kojima se pohranjuju varijable potrebne za izvršenje programa. Dijelimo ih na globalne podatkovne blokove (engl. *Global Data Block*) i pridružene podatkovne blokove (engl. *Instance Data block*).

Svakom pozivu funkcijskog bloka i funkcije dodjeljuju se ulazi i izlazi, parametri bloka. Na taj način, blok se može višekratno pozivati s istom funkcijom, ali različitim blokom parametara (lika 1.22).



Slika 1.22. Primjer poziva istog bloka s različitim parametrima

Nakon restarta, CPU ažurira ulazne i izlazne signale i poziva organizacijski blok OB 1.

Cikličko izvršavanje programa značajka je programabilnih kontrolera. Korisnički program također se izvršava ako se ne zahtijevaju nikakve akcije, primjerice, stroj ne radi. Uz ciklički izvođeni glavni program moguće je izvršiti i prekidom upravljano izvođenje programa. To može biti hardverski prekid, kao što je zahtjev za brzim odgovorom, ili ciklički prekid, događaj koji se odvija u definiranim intervalima.

Upravljački procesor prekida izvođenje glavnog programa kada se događaj dogodi i poziva dodijeljeni program prekida. Određenim događajima mogu se dodijeliti organizacijski blokovi koji se, u tom slučaju, obrađuju. Nakon što je prekidni program izvršen, upravljački procesor nastavlja s izvođenjem glavnog programa od točke prekida.

## 2. Varijable i tipovi podataka

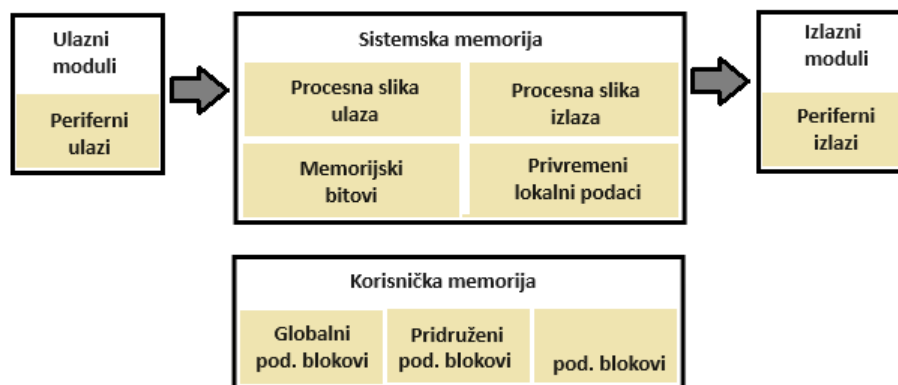
### 2.1. Operandi i tagovi

#### 2.1.1. Pojmovi

Obrada stanja ulaznih digitalnih i analognih signala odrađuje se na sljedeći način: signali se očitaju na ulazu, obrađuju u skladu s programom, a rezultati obrade proslijede se na digitalne i analogne izlaze.

Vrijednosti koje očitavamo, obrađujemo i prosljeđujemo koriste sljedeća memorijska područja (slika 2.1):

- periferne ulaze (engl. *Peripheral inputs*) - memorijsko područje na ulaznim modulima. Izravno sučelje s procesom ili strojem, očitavanje stanja sa senzora, preklopki, tastera
- procesnu sliku ulaza (engl. *Input process image*) - u sistemskoj memoriji, ulazna slika procesa
- periferne izlaze (engl. *Peripheral outputs*) - memorijsko područje na izlaznim modulima. Izravno sučelje s procesom ili strojem, postavljanje komandi prema signalizaciji, motorima, ventilima
- procesnu sliku izlaza (engl. *Output process image*) - u sistemskoj memoriji, izlazna slika procesa
- memorijske bitove (engl. *Bit memory*) - u sistemskoj memoriji
- privremene lokalne podatke (engl. *Temporary local data*) - u sistemskoj memoriji za privremenu pohranu podataka koji se obrađuju
- podatkovne blokove (engl. *Data blocks*) - memorijska područja u korisničkoj memoriji kojima se može pristupiti globalno iz svih dijelova programa (engl. *Global*), samo iz funkcijskog bloka (pridruženi podatkovni blokovi) ili lokalno, samo iz određene funkcije.



Slika 2.1. Memorijska područja za operande

Pristupanje stanju signala ili vrijednostima naziva se adresiranje.

Adresiranje može biti:

- **apsolutno** adresiranje

Koristi se operand koji se sastoji od ID operanda (tip podatka) i memorijske adrese

- primjerice, %I0.0 ID operanda u ovom je slučaju I što označava ulazni signal, adresa nulti (0.) bajt i bit 0
- primjerice, %Q4.0 ID operanda je izlazni signal, adresa 4. bajt, bit 0

npr. %M122.6 ID operanda je memorijski bajt, adresa 122. bajt, bit 6

- **simboličko** adresiranje

Primjerice, operandu I0.0 dodijeljeno je ime Senzor. Dodijeljeno ime operandu naziva se **Tag**. Ta imena ne smiju sadržavati navodnike, moraju biti jedinstvena unutar cijelog korisničkog programa.

ID operanda definira tip podatka koji je dodijeljen operandu, primjerice:

- Ix.x *BOOL* 1 bit, binarni operand ili binarni tag
- IBx *BYTE* 1 bajt, digitalni operand ili digitalni tag
- IWx *WORD* 16 bitova, 1 riječ, 2 bajta, digitalni operand ili digitalni tag
- IDx *DWORD* 32 bita, 1 dvostruka riječ, 4 bajta, digitalni operand ili digitalni tag.

Digitalni operandi ili tagovi mogu se razlikovati, ne samo po broju bitova koje sadrže, već i po vrsti:

- INT cijeli broj (*integer*) 16 bita
- CHAR znakovi u ASCII kôdu
- ARRAY polje podataka, nekoliko tagova istog formata, kao jedan tag.

### 2.1.2. Periferni ulazi i izlazi, procesni ulazi i izlazi

**Periferni ulazi** (engl. *Peripheral inputs*) operandi su na ulaznim modulima. Radi se o digitalnim ili analognim signalima koja putem ožičenja dolaze od procesa ili stroja. Ti signali prosljeđuju se, prije svakog ciklusa obrade korisničkog programa, u područje procesnih ulaza koje se nalazi u sistemskoj memoriji CPU-a.

**Periferni izlazi** (engl. *Peripheral outputs*) operandi su na izlaznim modulima. Radi se o digitalnim ili analognim signalima koji se putem ožičenja povezuju sa strojem ili procesom. Ti signali prosljeđuju se, nakon svakog ciklusa obrade korisničkog programa, iz područja procesnih izlaza koje se nalazi u sistemskoj memoriji CPU-a.

Svi moduli, ulazni ili izlazni, s centralnog PLC-a ili izdvojenih jedinica adresiraju se u jedinstvenom adresnom području. O vrsti modula ovisi veličina adresnog područja koje će zauzeti pa tako:

- modul od 8 digitalnih ulaza ili izlaza zauzima 1 bajt adresnog područja
- modul od 16 digitalnih ulaza ili izlaza zauzima 2 bajta adresnog područja
- modul od 32 digitalna ulaza ili izlaza zauzima 4 bajta adresnog područja
- modul od 2 analogna ulaza ili izlaza zauzima 4 bajta adresnog područja
- modul od 8 analognih ulaza ili izlaza zauzima 16 bajtova adresnog područja.

### 2.1.3. Memorijski bitovi

Memorijski bitovi smješteni su u sistemskoj memoriji CPU-a i uvijek im se može pristupiti. Najčešće se koriste za spremanje binarnih stanja tijekom obrade programa. Mogu se čitati i zapisivati bez ikakvog ograničenja.

**Retentivni memorijski bitovi** (engl. *Retentive bit memories*) oni su memorijski bitovi koji zadržavaju svoje stanje čak i nakon prestanka napajanja. Područje, unutar kojeg se nalaze, definira korisnik unutar tablice tagova.

**Sistemske memorijski bitovi** (engl. *System memory bits*) nalaze se na programiranom bajtu, primjerice, 0. Taj memorijski bajt kontrolira operativni sustav CPU-a:

- M0.0 (*First Scan*)  
postavlja se inicijalno, aktivan je samo za vrijeme prvog ciklusa kada ima stanje 1, inače je u stanju 0
- M0.1 (*DiagStatusUpdate*)  
postavlja se u stanje 1 ako je došlo do promjene u statusu izvođenja programa u odnosu na prethodni ciklus, za vrijeme STARTUP-a i prvog ciklusa je uvijek u stanju 0
- M0.2 (*Always 1*)  
uvijek je aktivan, u stanju 1, koristi se u korisničkom programu kao binarna konstanta
- M0.3 (*Always 0*)  
uvijek je pasivan, u stanju 0, koristi se u korisničkom programu kao binarna konstanta

**Clock memorijski bitovi** (engl. *Clock memory bits*) nalaze se na programiranom bajtu, primjerice, 1. Mnogi procesi unutar PLC-a zahtijevaju periodički signal točno određene frekvencije (slika 2.2).

Memorijski bit	7	6	5	4	3	2	1	0
Trajanje perioda (s)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Frekvencija (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Slika 2.2. Clock memorijski bajt

Vrijeme trajanja signala i stanke je u omjeru 1:1. Ako je programirano da je clock memorijski bajt na adresi 1, to bi značilo da na adresi %M1.3 imamo signal frekvencije 2Hz, to je signal koji 0.5 sekundi ima stanje 1, a sljedećih 0.5 sekundi stanje 0.

#### 2.1.4. Podatkovni blokovi

Podatkovni blokovi nalaze se unutar korisničke memorije. Koriste se sljedeće vrste podatkovnih blokova:

- globalni podatkovni blokovi (engl. *Global data blocks*)
- pridruženi podatkovni blokovi (engl. *Instance data blocks*). Pozivaju se iz funkcijskih blokova i njihova struktura tamo je definirana
- tipski blokovi podataka (engl. *Type data blocks*).

Podatkovne blokove adresiramo:

- apsolutno, koristeći se dodijeljenim im brojem
- simbolički, koristeći se njihovim imenom. Naziv bloka mora biti jedinstven unutar korisničkog programa.

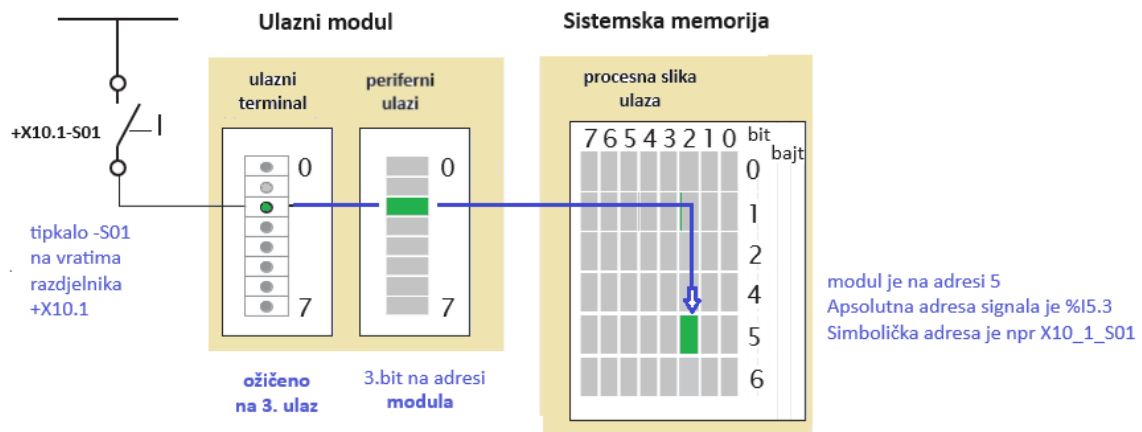
Podatci u podatkovnim blokovima mogu biti deklarirani podatci koji zadržavaju svoju vrijednost nakon nestanka napajanja (engl. *retentive*).

## 2.2. Adresiranje

### 2.2.1. Putanja signala

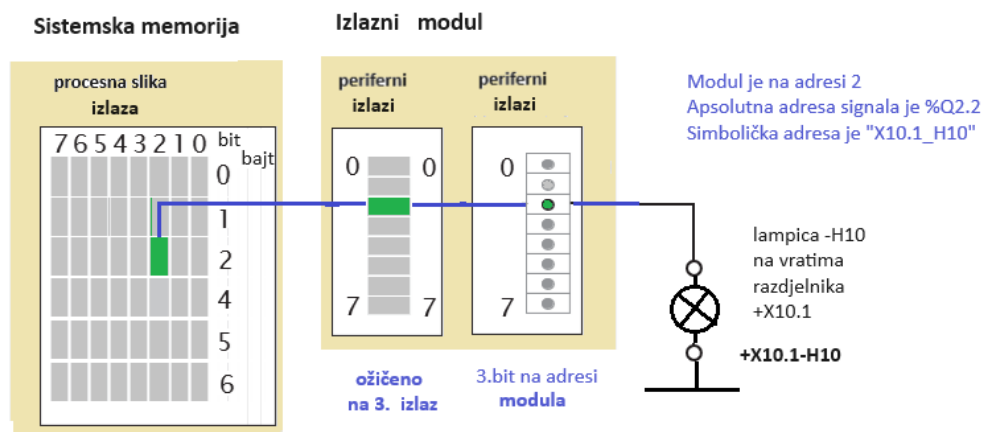
Ožičenje definirano elektroprojektom jednoznačno označava signale. Dobro je tom se logikom koristiti i u adresiranju u korisničkom programu radi lakšeg snalaženja.

Primjerice, tipkalo -S01 na vratima razdjelnika +X10.1 priključeno je na modul digitalnih ulaza na PLC-u (slika 2.3). U hardverskoj konfiguraciji dodjeljuje se adresa modula, a jednoznačno su određeni i modulski kanali. Adresa modula je, primjerice, %IB5, a signal je doveden na treći modulski kanal. Apsolutna adresa u PLC-u tog signala je %I5.2. Ova adresa istovremeno je i adresa na procesnoj slici ulaza. Tom operandu, digitalnom ulazu, može se dodijeliti simbolička adresa „Start motora“, a može i „X10\_1\_S01“.



Slika 2.3. Putanja signala od tipkala do procesne slike ulaza

U skladu s time je i putanja od procesne slike izlaza do ožičene lampice na vratima razdjelnika (slika 2.4). Apsolutna adresa u PLC-u tog signala je %Q2.2. Ta adresa je istovremeno i adresa na procesnoj slici izlaza. Tom operandu, digitalnom izlazu, može se dodijeliti simbolička adresa „Nalog za start“, a može i „X10\_1-H10“.



Slika 2.4. Putanja signala od procesne slike izlaza do lampice

## 2.2.2. Apsolutno adresiranje operanda

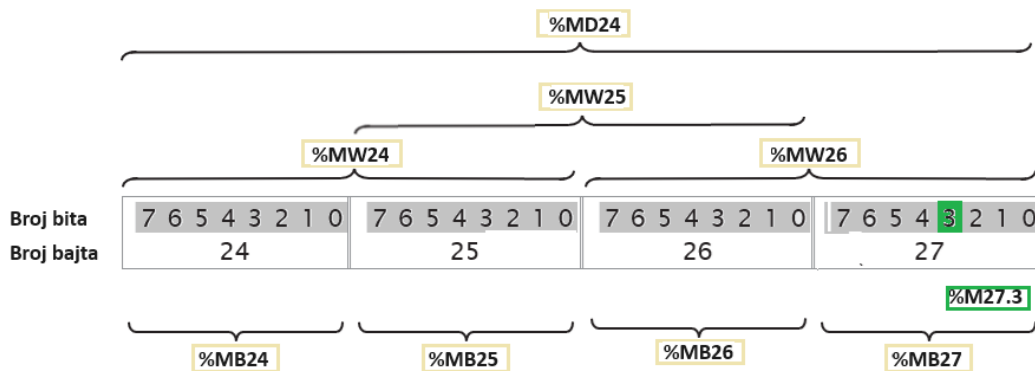
Kod apsolutnog adresiranja stanje signala kod digitalnih signala ili numerička vrijednost kod analognih signala adresira se izravno korištenjem adrese u području operanda. Operand, na primjer %I2.5, sadrži ID operanda, adresu bajta i adresu bita odvojenu točkom (tablica 2.1). Apsolutna adresa prikazuje se znakom postotka ispred (%).

Tablica 2.1. ID operanda i apsolutno adresiranje

	ID operanda	Bit (Bit) 1 bit	Bajt (Byte) 8 bitova	Riječ (Word) 16 bitova	Dvostruka riječ (Double word) 32 bita
Ulazi	<b>I</b>	<b>%Iy.x</b>	<b>%IBy</b>	<b>%IWy</b>	<b>%IDy</b>
Periferni ulazi	<b>P</b> se dodaje ulaznom operandu	<b>%Iy.x:P</b>	<b>%IBy:P</b>	<b>%IWy:P</b>	<b>%IDy:P</b>
Izlazi	<b>O</b>	<b>%Oy.x</b>	<b>%OBy</b>	<b>%OWy</b>	<b>%ODy</b>
Periferni izlazi	<b>P</b> se dodaje izlaznom operandu	<b>%Oy.x:P</b>	<b>%OBy:P</b>	<b>%OWy:P</b>	<b>%ODy:P</b>
Memorijski bit	<b>M</b>	<b>%My.x</b>	<b>%MBy</b>	<b>%MWy</b>	<b>%MDy</b>
Podatkovni blokovi	<b>DB</b>	<b>%DBz.DBXy.x</b>	<b>%DBz.DBBy</b>	<b>%DBz.DBWy</b>	<b>%DBz.DBDy</b>
Privremeni lokalni podatci	<b>L</b>	Apsolutno adresiranje nije moguće			

z= broj podatkovnog bloka, y=adresa bajta, x=adresa bita

Bitovi u bajtu broje se sdesna na lijevo, počevši od nule. Brojanje počinje od početka za svaki bajt. Svako područje operanda organizirano je u bajtovima. Bajtovi se broje počevši od početka područja s nulom. S operandom veličine bajta, broj bajta naveden je kao adresa bajta; s operandom veličine riječi, broj najmanjeg bajta naveden je kao adresa bajta, a s operandom veličine dvostruke riječi, broj najmanje značajnog bajta u dvostrukoj riječi je adresa operanda (slika 2.5).



Slika 2.5. Raspored bitova, bajtova, riječi i dvostrukih riječi

### 2.2.3. Simbolično adresiranje

Operandu se dodjeljuje alfanumerički ID (ime, simbol) i tip podatka. ID operanda i ip podatka čine Tag (oznaka).

Primjerice, operand %I27.3 imena Taster je tip podatka BOOL. Tag Taster u korisničkom programu može se koristiti umjesto operanda %I27.3.

Nazivi tagova mogu se sastojati od slova, znamenki i posebnih znakova (osim dvostrukih navodnika). Pritom, ne pravi se razlika između velikih i malih slova.

Drugim riječima, sasvim je svejedno je li naziv Taga taster ili TASTER.

### 2.2.4. Adresiranje konstanti

Konstanta je fiksna brojučana vrijednost. Raspon dopuštenih vrijednosti ovisi o tipu podatka (tablica 2.2).

Tablica 2.2. Raspon dopuštenih vrijednosti konstanti

<b>Bit</b> (Bit) 1 bit	<b>Bajt</b> (Byte) 8 bitova	<b>Riječ</b> (Word) 16 bitova	<b>Dvostruka riječ</b> (Double word) 32 bita
(0, 1)	(B#16#00 ... B#16#FF)	(W#16#0000... W#16#FFFF)	(DW#16#0000 0000 ... DW#16#FFFF FFFF)

Konstante koje nisu cjelobrojni brojevi, mogu se unositi i kao:

- decimalni format (primjerice, 123,456)
- eksponencijalni format (*floating point*) (primjerice, +1,23456E+02 isto što i +12,3456E+01).

I konstante mogu imati dodijeljena jedinstvena imena.

### 2.2.5. Indirektno adresiranje

Indirektno ili neizravno adresiranje koristi se kada se isti programski dijelovi koriste višestruko, ali s različitim operandima. Tada je vrlo teško pratiti rad programa i zahtijeva značajnu vještinu programera.

## 2.3. Opće informacije o tipovima podataka

Tipovi podataka definiraju svojstva tagova i dopušteni raspon njegovih vrijednosti.

Razlikuju se:

- elementarni podatci, veličine 1-64 bita
- strukturirani podatci, kombinacija elementarnih tipova podataka
- parametri, prenošenje parametara kroz funkcije i funkcijski blokovi
- strukture koje definira korisnik
- sistemski podatci fiksne strukture
- podatci definirani hardverskom konfiguracijom.

Kada se podatci uspoređuju ili zbrajaju, ili kada se uvoze u blok, uključeni tagovi moraju imati isti ili usporediv tip podataka.

Na primjer, tag s formatom podataka DWORD (izvorni tip podataka) ne može se primijeniti na parametar bloka koji očekuje tip podataka WORD (određeni tip podataka). Programirana duljina bita mora odgovarati očekivanoj duljini bita za parametar *in-out*.

Tipovi podataka tagova mogu se pretvoriti u odgovarajući tip. To se može dogoditi automatski s implicitnom konverzijom tipa podataka ili s funkcijama za eksplicitnu konverziju tipa podataka.

U ovom priručniku pojašnjeni su elementarni podatci i sistemski podatci fiksne strukture.

## 2.4. Elementarni tipovi podataka

Elementarni tipovi podataka veličine su 1, 8, 16, 32 ili 64 bita. Tipovi podataka BCD16 i BCD32 nisu tipovi podataka u užem smislu – ne mogu se dodijeliti oznaci. Relevantni su samo za pretvorbu tipa podataka.

### 2.4.1. BOOL, BYTE, WORD and DWORD

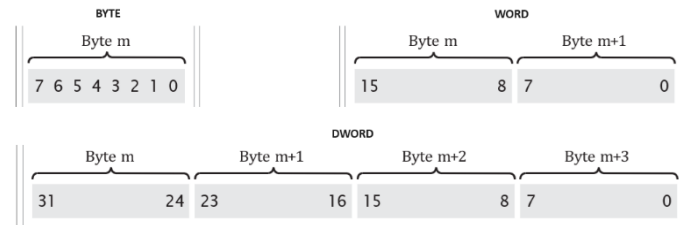
**BOOL** je tip podatka koji predstavlja vrijednost jednog bita. Tag može imati vrijednost 0 ili 1 ili biti u stanju FALSE ili TRUE .

**BYTE** je veličine 8 bita. Konstante su u rasponu od 16#00 do 16#FF.

**WORD** je veličine 16 bitova. Konstante su u rasponu od 16#0000 do 16#FFFF. Konstanta se može zapisati i u obliku 16-bitnog binarnog broja u rasponu od 2#0000\_0000\_0000\_0000 to 2#1111\_1111\_1111\_1111).

**DWORD** je veličine 32 bita.

Konstante su u rasponu od 16#0000\_0000 do 16#FFFF\_FFFF. Konstanta se može zapisati i u obliku 32-bitnog binarnog broja u rasponu od 2#0000\_0000\_0000\_0000\_0000\_0000\_0000\_0000 do 2#1111\_1111\_1111\_1111\_1111\_1111\_1111\_1111 (slika 2.6).



Slika 2.6. Struktura BYTE, WORD i DWORD

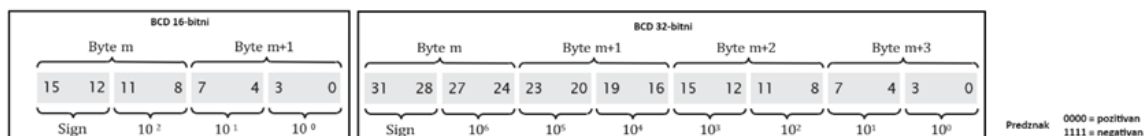
### 2.4.2. BCD-kodirani brojevi

BCD-kodirani brojevi rabe WORD ili DWORD (lika 2.7).

U slučaju 16-bitnog BCD broja, na bitu 15 nalazi se najznačajniji bit koji mu određuje predznak - 0 znači da je broj pozitivan, a 1 da je broj negativan. Numerički raspon za 16-bitne brojeve je od 0 do  $\pm 999$ .

U slučaju 32-bitnog BCD broja, na bitu 31 nalazi se najznačajniji bit koji mu određuje predznak, 0 znači da je broj pozitivan, a 1 da je broj negativan. Numerički raspon za 32-bitne brojeve je od 0 do

$\pm 9\,999\,999$ .



Slika 2.7. Struktura za BCD 16-bitni i BCD 32-bitni

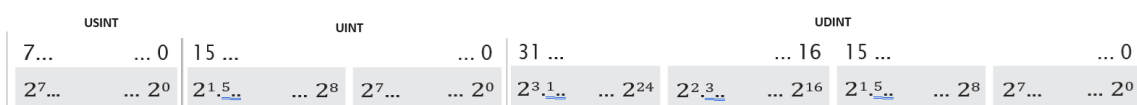
### 2.4.3. USINT, UINT i UDINT, te SINT, INT and DINT

USINT, UINT i UDINT veličine su bez predznaka (slika 2.8):

**USINT**, veličine 1 BYTE, raspon brojeva od 0 do 255

**UINT**, veličine 1 WORD, raspon brojeva od 0 do 65535

**UDINT**, veličine 1 DWORD, raspon brojeva od 0 do 4 294 967 295



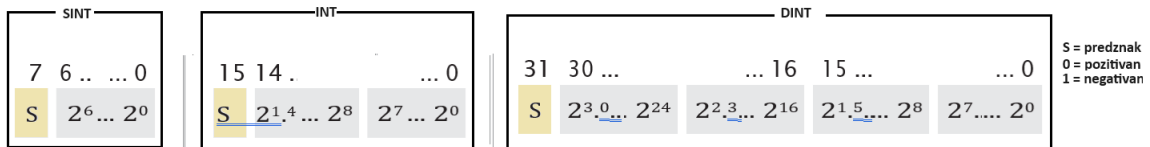
Slika 2.8. Struktura za USINT, UINT i UDINT

SINT, INT i DINT veličine su bez predznaka (slika 2.9):

**SINT**, s predznakom, veličine BYTE, raspon brojeva od -256 do +255

**INT**, s predznakom, veličine 1 WORD, raspon brojeva od -32 768 do +32 767

**DINT**, s predznakom, veličine 1 DWORD, raspon brojeva od -2147483648 do +2147483647



Slika 2.9. Struktura za SINT, INT i DINT

### 2.4.4. REAL i LREAL

**REAL**, 32-bitni broj, sastoji se od 3 komponente: predznaka, 8-bitnog eksponenta i 23-bitne mantise.

Raspon broja je:

- za negativne brojeve od  $-3,402\ 823 \times 10^{+38}$  do  $-1.175\ 494 \times 10^{-38}$
- za pozitivne brojeve od  $+1.175\ 494 \times 10^{-38}$  do  $+3.402\ 823 \times 10^{+38}$ .



Slika 2.10. Struktura za REAL i LREAL

**LREAL**, 64-bitni broj, sastoji se od 3 komponente: predznaka, 11-bitnog eksponenta i 52-bitne mantise

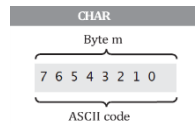
Raspon broja je:

- za negativne brojeve  
 $-1,797\ 693\ 134\ 862\ 3158 \times 10^{+308}$  do  $-2.225\ 073\ 858\ 507\ 2014 \times 10^{-308}$
- za pozitivne brojeve  
 $+2.225\ 073\ 858\ 507\ 2014 \times 10^{-308}$  to  $+1.797\ 693\ 134\ 862\ 3158 \times 10^{+308}$

### 2.4.5. CHAR

Tag tipa podatka CHAR (znak) zauzima jedan bajt.

Tip podataka CHAR predstavlja jedan znak koji se sprema u ASCII formatu.



Slika 2.11. Struktura za CHAR

## 2.5. Sistemski podatci fiksne strukture

Sistemski tipovi podataka (SDT) unaprijed su definirani tipovi podataka koji se sastoje od fiksnog broja komponenti od kojih svaka ima različite osnovne tipove podataka. Sistemski tipovi podataka isporučuju se zajedno s alatom za programiranje i ne mogu se mijenjati.

Sistemski tipovi podataka mogu se koristiti samo zajedno s određenim funkcijama ili izrazima.

Sistemski tipovi podataka koji se ovdje upotrebljavaju su:

- IEC tajmeri
- IEC brojači.

### 2.5.1. IEC tajmeri

Podatci pojedine naredbe IEC tajmera strukturirani su u skladu sa strukturom sistemskog tipa podataka IEC\_TIMER. Ako se koriste naredbe TP, TON, TOF ili TONR, programski alat, ovisno o specifikaciji, kreira podatkovni blok.

Za sve tajmere podatkovni blok sadrži sljedeće parametre - IEC\_TIMER struktura:

- IN (engl. INput) - signal na ulazu
- PT (engl. **P**reset **T**imer) - trajanje tajmera, mora biti pozitivna vrijednost
- Q (engl. *output*) - signal na izlazu
- ET (engl. **E**lapsed **T**ime) - trenutna vrijednost tajmera

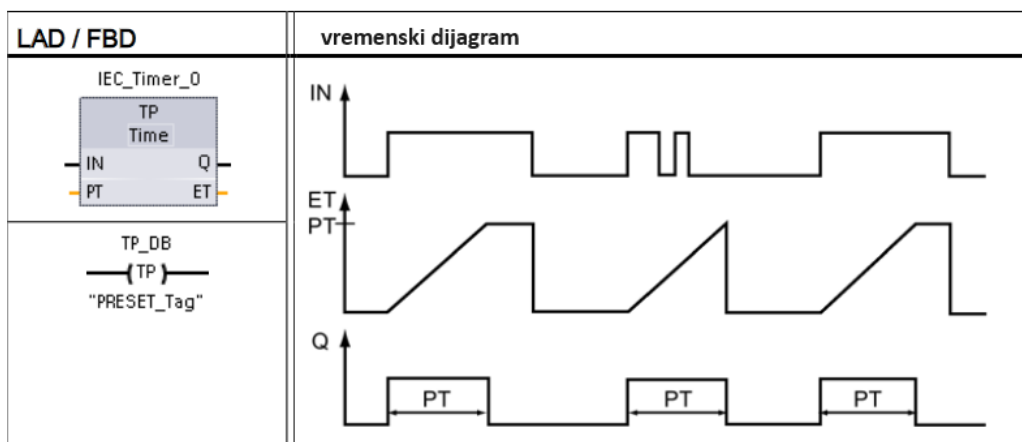
Vrijeme (engl. *TIME*) prikazuje se u milisekundama.

Opseg vrijednosti je od T#-24d20h31m23s648ms do T#+24d20h31m23s64ms

Tablica 2.3. IEC tajmeri

Vremenska funkcija	Tip podatka	IEC tip podatka
TP- Pulsni tajmer	TIME	TP_TIME
TON- Tajmer po uključenju	TIME	TON_TIME
TOFF- Tajmer po isključenju	TIME	TOF_TIME
TONR- Retentivni tajmer	TIME	TONR_TIME

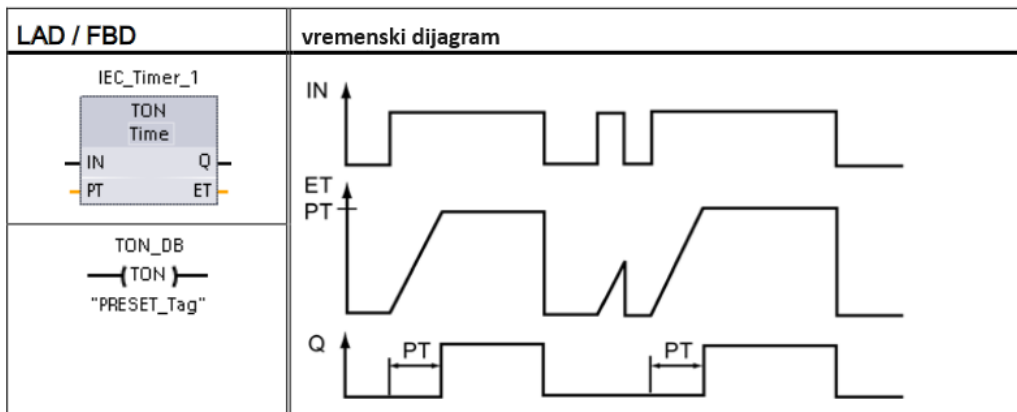
**Pulsni tajmer TP** (engl. *Timer Pulse*)



Slika 2.12. Pulsni tajmer

Tajmer TP generira, nakon promjene IN ulaza iz 0 u 1 Q impuls. Impuls traje PT vremena. PT vrijeme definirano je vrijednošću PT parametra. Parametar Time jednak je PTx1mS.

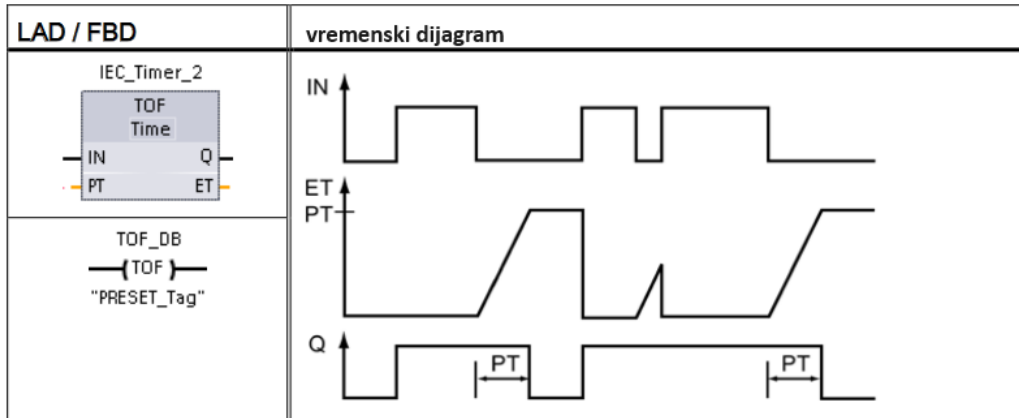
**Tajmer po uključenju TON** (engl. *Timer ON delay*)



Slika 2.13. Tajmer po uključenju

Tajmer TON postavlja izlaz Q s kašnjenjem u odnosu na IN ulaz. Kašnjenje iznosi TP vremena. Izlaz Q prelazi u nisko stanje kada ulaz IN prijeđe u nisko stanje.

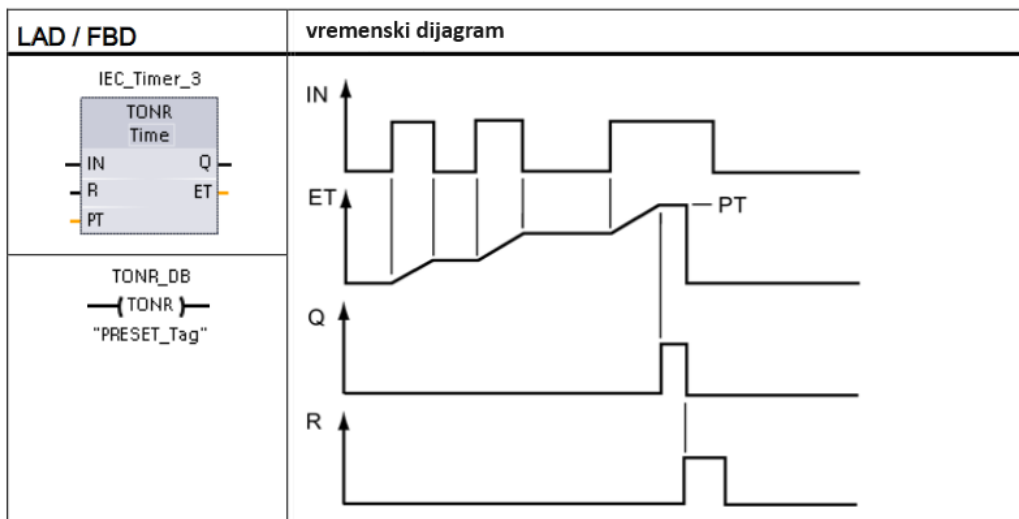
### Tajmer po isključenju TOF (engl. *Timer OFF delay*)



Slika 2.14. Tajmer po isključenju

Tajmer TOF postavlja izlaz Q u visoko stanje kada je ulaz IN u visokom stanju. Izlaz Q postavljen je u visoko stanje dulje od visokog stanja na IN ulazu. Kada ulaz IN prijeđe u nisko stanje, izlaz Q zadržava visoko stanje još TP vremena.

### Akumulirajući tajmer TONR (engl. *Timer ON delay retentive*)



Slika 2.15. Retentivni tajmer

Tajmer TONR postavlja izlaz Q u odnosu na ulaz IN. Kada na ulazu IN u tajmer visoko stanje dosegne PT vremena, izlaz Q kratkotrajno prelazi u visoko stanje. Tajmer se, nakon toga, resetira na nulu.

### 2.5.2. IEC brojači

Podatci funkcije IEC brojača strukturirani su ovisno o vrsti brojača u skladu sa sistemskim tipovima podataka C\_SCOUNTER (SINT vrsta podataka), IEC\_COUNTER (INT), IEC\_DCOUNTER (DINT), IEC\_US\_COUNTER (USINT), IEC\_UCOUNTER (UINT ) i IEC\_UDCOUNTER (UDINT).

Tablica 2.1. ID operanda i apsolutno adresiranje

Tip brojača	Tipovi podataka	IEC tipovi podataka
Up counter	SINT INT DINT	CTU_SINT CTU_INT CTU_DINT
Down counter	SINT INT DINT	CTD_SINT CTD_INT CTD_DINT
Up/down counter	SINT INT DINT	CTUD_SINT CTUD_INT CTUD_DINT

Ukupan broj brojača unutar korisničkog programa ograničen je količinom memorije. Svaki brojač koristi se strukturom u pripadajućem podatkovnom bloku DB (slika 2.16).

Name	Data type	
Static		
CU	Bool	} 2 bajta
CD	Bool	
R	Bool	
LD	Bool	
QU	Bool	
QD	Bool	
PV	Int	2 bajta
CV	Int	2 bajta

Slika 2.16. IEC brojač DB

Brojač kojem je format podataka:

- SINT, za prikaz podatka upotrebljava 3 bajta
- INT, za prikaz podatka upotrebljava 6 bajtova
- DINT, za prikaz podatka upotrebljava 12 bajtova.

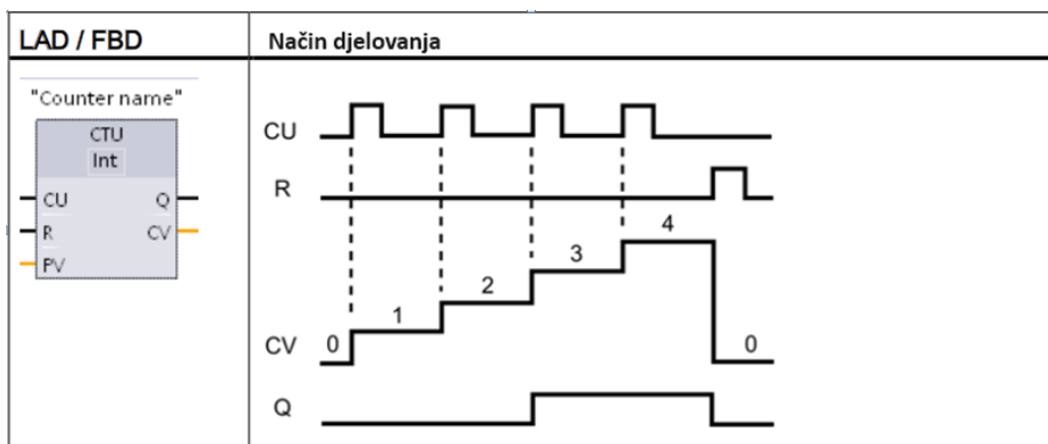
Tipovi brojača su:

- brojač naviše
- brojač naniže
- brojač naviše/naniže.

### Brojač naviše CTU (engl. *Counter Up*)

Svaki put kada na ulazu u brojač CTU (engl. *Counter Up*) signal CU (engl. *Count Up input*) prijeđe iz niskog u visoko stanje, trenutna vrijednost brojača CV (engl. *Current count Value*) poveća se za jedan. Brojanje započinje od nule i može ići do zadane vrijednosti PV (engl. *Preset Value*). Kada trenutna vrijednost brojača dostigne zadanu vrijednost, postavlja se izlaz Q u visoko stanje i ostaje u njemu. Signalom reseta R (engl. *Reset*) brojač CTU vraća se na 0, a izlaz Q postavlja u nisku vrijednost. U primjeru na slici 2.15 PV=3, izlaz Q je u stanju 1 dok je CV jednak i veći od PV.

Brojači tipa CTU obično se koriste za praćenje koliko se puta nešto dogodilo.

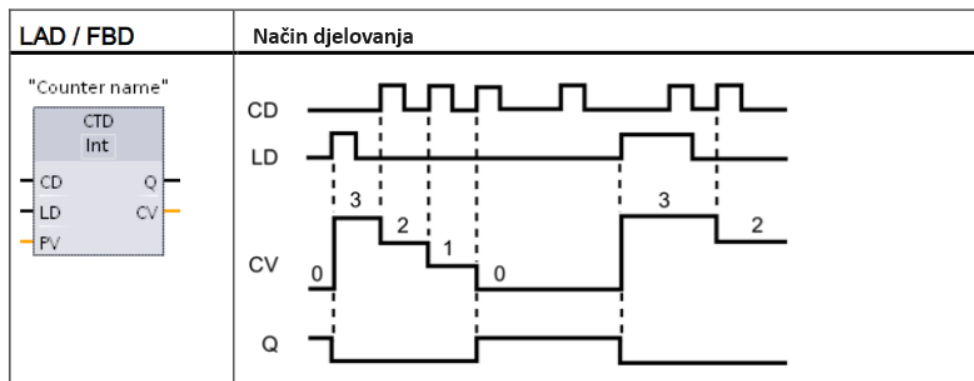


Slika 2.17. Brojač CTU

### Brojač naniže CTD (engl. *Counter Down*)

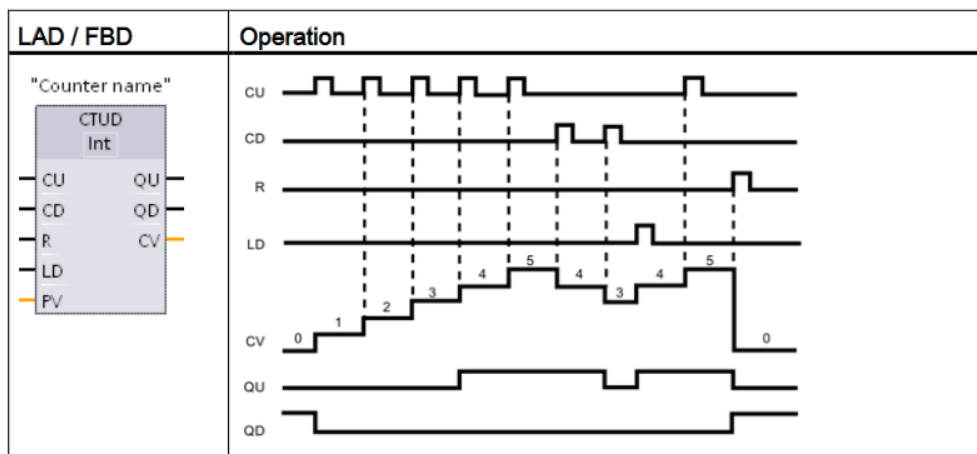
Svaki put kada na ulazu u brojač CTD (engl. *Counter Down*) signal CD (engl. *Count Down input*) prijeđe iz niskog u visoko stanje, trenutna vrijednost brojača CV (engl. *Current count Value*) snižava se za 1. Brojanje započinje od vrijednosti PV (engl. *Preset Value*) i ide do 0. Kada trenutna vrijednost brojača postane vrijednost 0, postavlja se izlaz Q u visoko stanje. Kada vrijednost na ulazu LD (engl. *Load input*) prijeđe iz niskog u visoko stanje, vrijednost PV postavlja se u brojač kao nova vrijednost CV.

U ovom primjeru PV=3, kad se zadovolji uvjet CV= 0, izlaz Q prelazi u stanje 1, a kada signal LD prijeđe iz niskog u visoko stanje, u parametar CV upisuje se PV, CV=PV.



Slika 2.18. Brojač CTD

**Brojač naviše/naniže CTUD** (engl. **C**ounter **U**p and **D**own)



Slika 2.19. Brojač CTUD

Svaki put kada na ulazu u brojač CTUD (engl. **C**ounter **U**p and **D**own) signal CU (engl. **C**ount **U**p input) prijeđe iz niskog u visoko stanje, trenutna vrijednost brojača CV (engl. **C**urrent count **V**alue) poveća se za 1. Brojanje započinje od nule i može ići do zadane vrijednosti PV (engl. **P**resent **V**alue). Kada trenutna vrijednost brojača dostigne zadanu vrijednost, postavlja se izlaz QU u visoko stanje i ostaje u njemu. Signalom reseta R (engl. **R**eset) brojač CTU vraća se na 0, a izlaz QU postavlja se u nisku vrijednost.

Svaki put kada na ulazu u brojač CTUD (engl. **C**ounter **D**own) signal CD (engl. **C**ount **D**own input) prijeđe iz niskog u visoko stanje, trenutna vrijednost brojača CV (engl. **C**urrent count **V**alue) snižava se za 1. Kada trenutna vrijednost brojača postane vrijednost 0, izlaz QD postavlja se u visoko stanje. Kada vrijednost na ulazu LD (engl. **L**oad input) prijeđe iz niskog u visoko stanje, vrijednost PV postavlja se u brojač kao nova CV.

Npr. u ovom primjeru PV=4, izlaz QU je u visokom stanju dok je CV jednak i veći od PV. Kada je CV=0, izlaz QD je u niskom stanju, a kad signal LD prijeđe iz niskog u visoko stanje, parametar CV poprima vrijednost parametra PV, CV=PV.

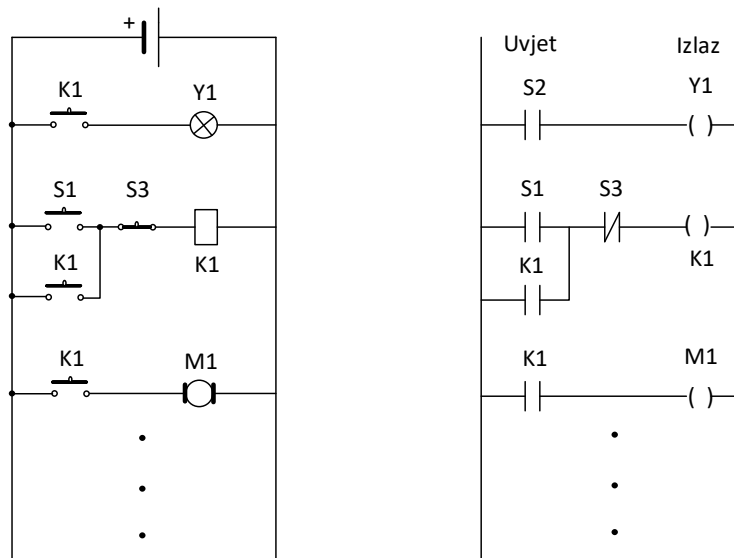
## 3. Ljestvičasti dijagrami - programiranje

### 3.1. Uvod

Prije programabilnih logičkih kontrolera, proizvodni pogoni koristili su se strujnim krugovima temeljenim na relejima za napajanje različitih izvršnih komponenti na temelju načina na koji su releji međusobno povezani. Nacrta električnih krugova - korišteni za realizaciju sustava upravljanja - poznati su kao ljestvičasti dijagrami. Releji su bili skupi, zahtijevali su stalno održavanje i nije ih bilo lako rekonfigurirati. Kako su PLC-ovi preuzeli taj proces, bilo je bitno zadržati sličnost starog sustava. Tako je stvorena ljestvičasta logika kao prvi PLC programski jezik.

Među nekoliko programskih jezika ljestvičasti logički dijagram osnovni je i najjednostavniji programski jezik koji služi programiranju PLC-a. Programski jezik ljestvičastih dijagrama grafički je programski jezik.

Donja slika prikazuje električnu shemu upravljanja svjetiljkom i motorom pomoću dviju sklopki, S1 i S3:



Slika 3.1. Ljestvičaste sheme

U slučaju da se sklopka S1 zatvori, aktivira se sklopnik K1 kao i njemu pridruženi kontakti. Preko kontakata K1 uključit će se svjetiljka Y1 i motor M1. Ovdje se tri vodoravne linije nazivaju prečkama koje su povezane između dviju okomitih linija koje se nazivaju sabirnice. Svaka prečka uspostavlja električni kontinuitet između pozitivne (+) i negativne sabirnice (-) tako da struja teče od ulaznih prema izlaznim uređajima. Neki od simbola koji se koriste u programiranju ljestvičaste logike, prikazani su na slici. Pored električne sheme prikazan je i njezin grafički ekvivalent realiziran pomoću grafičkih simbola.

Ljestvičasti dijagram tako je nazvan jer softverske naredbe nalikuju na ljestve. Na lijevoj strani ljestvičaste sheme nalaze se uvjeti, dok se na desnoj strani nalaze izlazi koji se pokreću ako su uvjeti s lijeve strane ispunjeni. Svaka prečka ljestvice proteže se slijeva na desno i PLC je izvršava od vrha prema dolje.

Programska naredba izvršava se u nekoliko koraka:

- **korak 1** - hipotetska struja počinje se kretati slijeva na desno
- **korak 2** - kada hipotetska struja naiđe na naredbu K1, provjerava je li uvjet TRUE ili FALSE. Ako je K1 False, PLC prekida ovu prečku
- **korak 3** - hipotetska struja ide na sljedeću naredbu. Ponavlja korak 2 dok se prečka ne završi
- **korak 4** - PLC se pomiče na prečku ispod.

Prečke se u programskom jeziku nazivaju mreže (*engl. Network*).

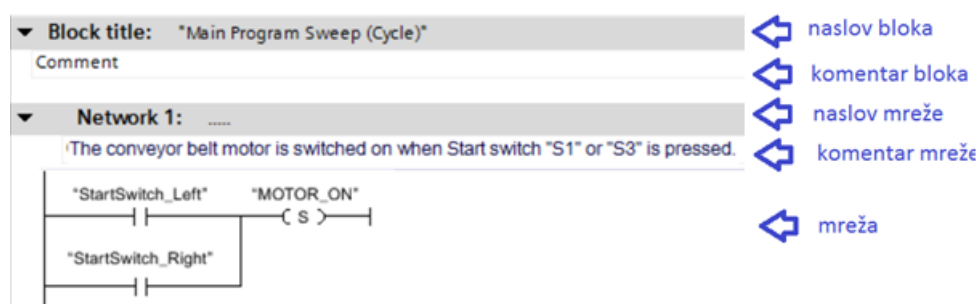
U ovom poglavlju opisuje se programiranje ljestvičastim dijagramima koristeći se nizom primjera.

### 3.1.1. Osnova programiranja ljestvičastim dijagramima

Korisnički program organiziran je pomoću:

- organizacijskih blokova OB (*engl. Organization Block*)
- funkcijskih blokova FB (*engl. Function Block*)
- funkcija FC (*engl. FunCtion*).

Blokovi su podijeljeni u dijelove koji se nazivaju mreže (*engl. Network*). Struktura programskog bloka s ljestvičastim dijagramom prikazana je na sljedećoj slici (slika 3.2).



Slika 3.2. Struktura programskog bloka

Naslov bloka (*engl. Block title*) i komentar bloka (*engl. Comment*), kao i naslov mreže i njen komentar nisu obavezni, ali ih je poželjno upotrebljavati.

### 3.1.2. Programske naredbe LAD

Tipovi programskih naredbi (Tablica 3.1.) kojima se koristimo kod programiranja ljestvičastim dijagramima su:

Tablica 3.1. Tipovi programskih naredbi LAD

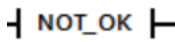
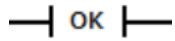
Kontakti	Binarne logičke funkcije	Standardni blokovi	Blokovi sa Q izlazom	Blokovi sa EN ulazom i ENO izlazom	Poziv podatkovnog bloka
<p><b>Binarni tag</b></p> <p><b>Binarni tag</b></p>					
NO i NC kontakti, impulsni kontakti, uvjetni kontakti	AND, OR i exclusive OR blok	pulsna funkcija	memorijske funkcije, tameri i brojači	matematičke funkcije	okvir predstavlja pozvani blok sa njegovim ulaznim i izlaznim parametrima

### 3.2. Programiranje relejnim kontaktima

Pri programiranju se moguće koristiti sljedećim kontaktima:

Tablica 3.2. Relejni kontakti

Tip	Opis
<p><b>Binarni tag</b></p>	normalno otvoren kontakt, NO
<p><b>Binarni tag</b></p>	normalno zatvoren kontakt, NC
<p><b>Binarni tag</b></p> <p><b>Triger tag</b></p>	pozitivni brid binarnog taga
<p><b>Binarni tag</b></p> <p><b>Triger tag</b></p>	negativni brid binarnog taga

Floating point tag 	invalidan tag
Floating point tag 	validan tag

### 3.2.1. NO i NC kontakt

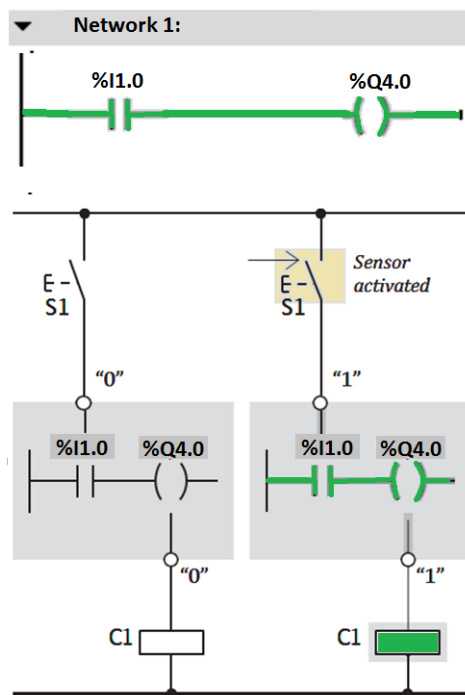
Primjeri jednostavnog programa i način prikupljanja podataka i upravljanja s izvršnim komponentama.

Sklopke i sklopnici priključuju se na ulaze i izlaze PLC-a električnim putem. Pojedini fizički ulazi povezuju se s programskim simbolima pomoću adresa. Ako se simbolu dodijeli adresa ulaza, tada će simbol biti aktivan ili neaktivan (uvjet ispunjen) ovisno o električnom stanju ulaza čija mu je adresa dodijeljena.

Slično vrijedi i za izlaze.

Aktivirat će se onaj kontaktor koji je spojen na izlaz PLC-a čija je adresa dodijeljena aktivnom simbolu za izlaz.

Način rada (slika 3.3 i slika 3.4):



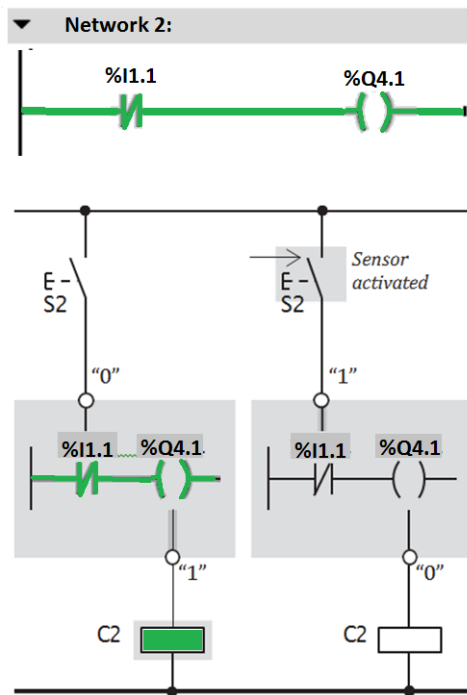
Slika 3.3. Princip rada NO kontakta

Network 1:

Ako je aktivan ulaz %I1.0 (sklopka S1), aktivira se i izlaz %Q4.0 (kontaktor C1).

Relejna shema 1:

Ako je sklopka S1 otvorena, C1 nije aktivan. Pritiskanjem sklopke S1 aktivira se kontaktor C1, tj. na njega se dovodi napon.



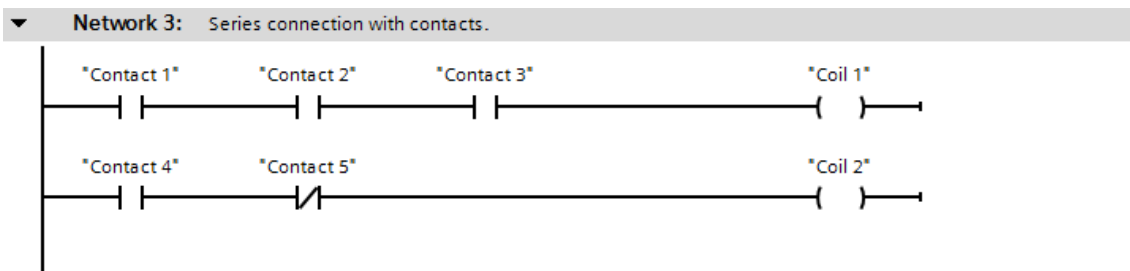
Slika 3.4. Princip rada NC kontakta

Network 2:

Ako nije aktivan ulaz %I1.1, (sklopka S1) aktivira se izlaz %Q4.1 (kontaktor C1).

Relejna shema 2:

Ako je sklopka S2 neaktivna, aktivira se kontaktor C2, tj. stavlja se pod napon. Ako je sklopka S2 aktivna, kontaktor C2 nije pod naponom.



Slika 3.5. Serijski spoj

Serijski spoj više binarnih tagova aktivira izlaznu vrijednost samo ako su svi normalno otvoreni binarni tagovi aktivni, a normalno zatvoreni binarni tagovi neaktivni.

Opis rada

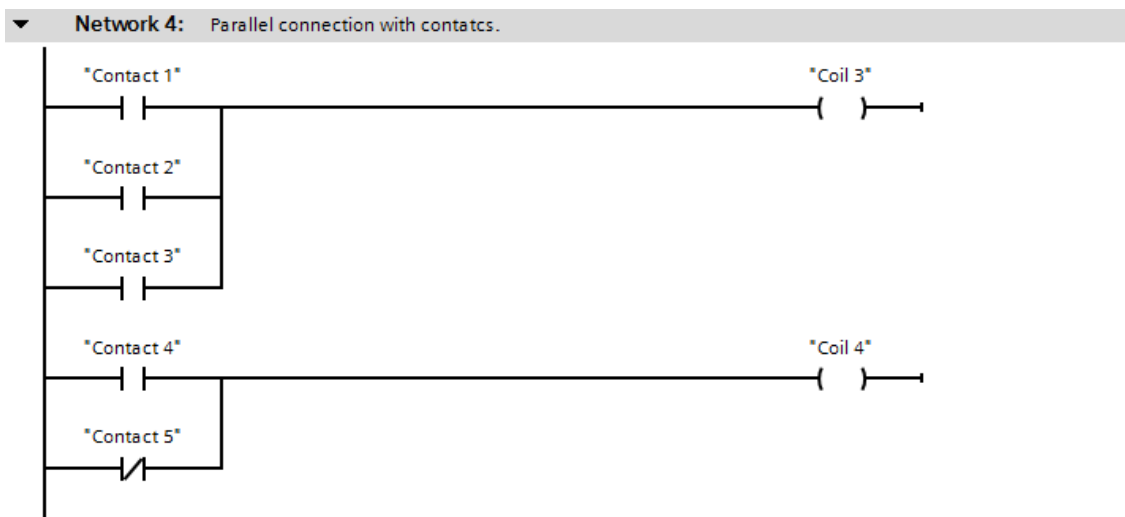
Network 3:

Samo ako su aktivni i Contact1 i Contact2 i Contact3, aktivira se Coil1.

Samo ako je aktivan Contact4 i neaktivan Contact5, aktivira se Coil2.

### 3.2.3. Paralelni spoj

Način rada (slika 3.6):



Slika 3.6. Paralelni spoj

Paralelni spoj više binarnih tagova aktivira izlaznu vrijednost ako je bar jedna grana tog paralelnog spoja aktivirana.

Opis rada

*Network 4:*

Ako je aktivna bar jedna grana ovoga spoja, odnosno bar jedan od Contact1 ili Contact2 ili Contact3, aktivira se Coil3.

Ako je aktivna bar jedna grana ovoga spoja, odnosno aktivan Contact4 ili neaktivan Contact5, aktivira se Coil4.

### 3.2.4. Serijsko-paralelni spoj

Serijsko-paralelni spoj više binarnih tagova aktivira izlaznu vrijednost ako je bar jedna grana tog paralelnog spoja aktivirana. Paralelna grana postaje aktivna ako su svi normalno otvoreni binarni tagovi aktivni, a normalno zatvoreni binarni tagovi neaktivni.

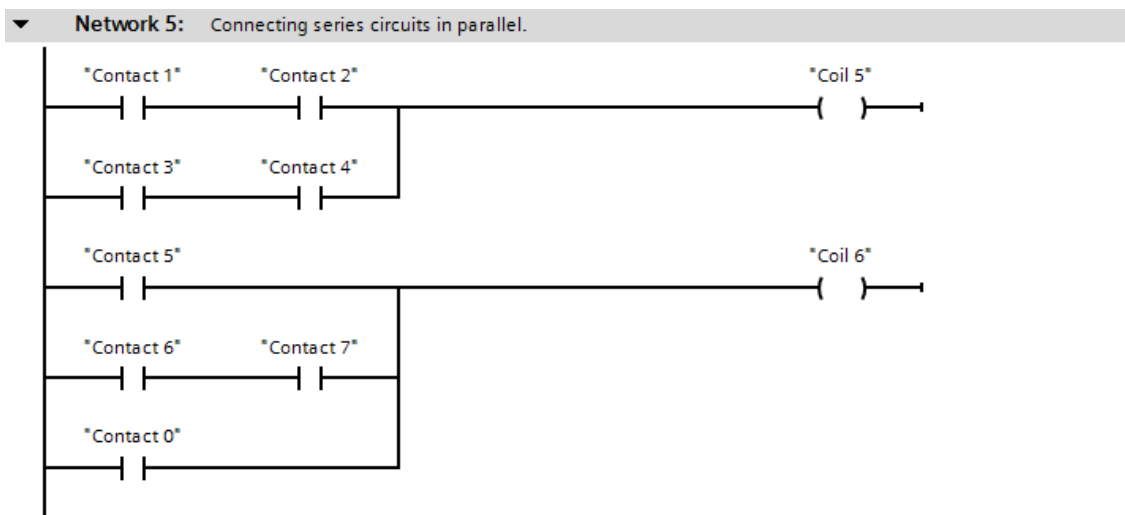
Način rada (slika 3.7)

Opis rada

*Network 5:*

U prvom ljestvičastom dijagramu: ako je aktivna bar jedna grana ovoga spoja - ako je to grana 1 i Contact1 i Contact2, ako je to grana 2 i Contact3 i Contact4 - aktivira se Coil5.

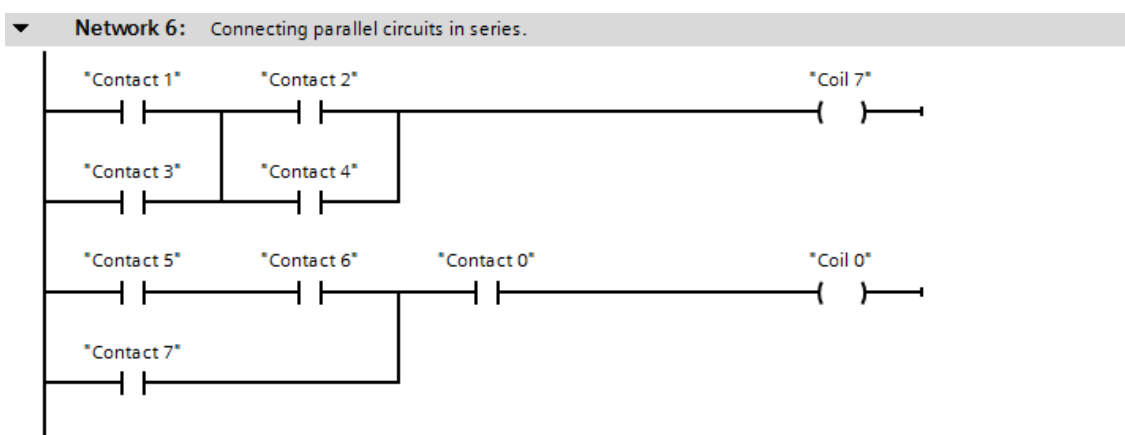
U drugom ljestvičastom dijagramu: ako je aktivna bar jedna grana ovoga spoja - ako je to grana 1 i Contact5, ili ako je to grana 2 i Contact6 i Contact7, ili ako je to grana 3, ako je aktivan Contact0, aktivira se Coil6.



Slika 3.7. Serijsko-paralelni spoj

### 3.2.5. Kombinirani serijsko-paralelni spoj

Način rada (slika 3.8):



Slika 3.8. Kombinirani serijsko-paralelni spoj

Opis rada

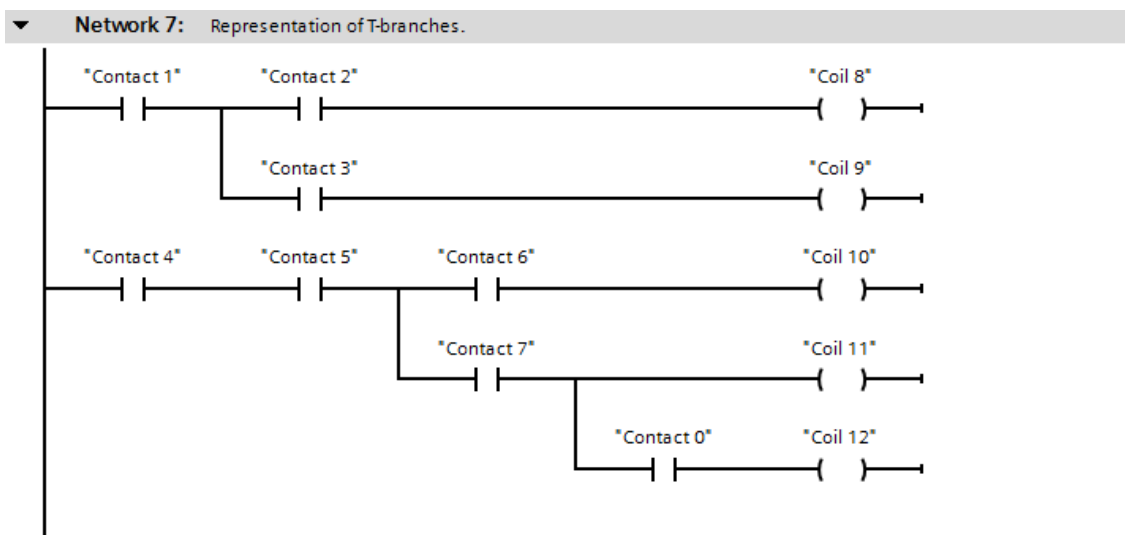
Network 6:

U prvom ljestvičastom dijagramu: ako je aktivan bar jedan Contact1 ili Contact3, i ako je aktivan bar jedan Contact2 ili Contact4, aktivira se Coil7.

U drugom ljestvičastom dijagramu: ako je aktivna bar jedna grana, ili grana 1 s aktivnim Contact5 i Contact6, ili grana 2 s aktivnim Contact7, te uz obavezno aktivan Contact0, aktivira se Coil6.

### 3.2.6. T-grananje

Način rada (slika 3.9):



Slika 3.9 T-grananje

Opis rada

*Network 7:*

U prvom ljestvičastom dijagramu:

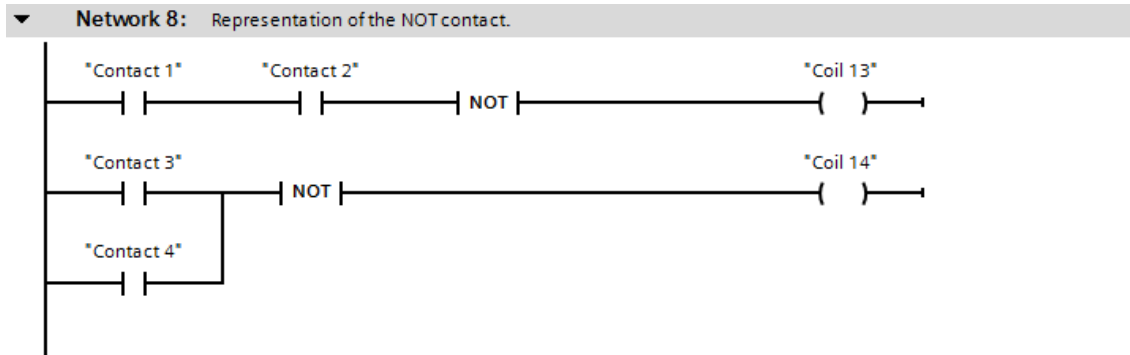
- aktivira se Coil8 samo ako su aktivni i Contact1 i Contact2
- aktivira se Coil9 samo ako su aktivni i Contact1 i Contact3

U drugom ljestvičastom dijagramu:

- aktivira se Coil10 samo ako su aktivni i Contact4 i Contact5 i Contact6
- aktivira se Coil11 samo ako su aktivni i Contact4 i Contact5 i Contact7
- aktivira se Coil12 samo ako su aktivni i Contact4 i Contact5 i Contact7 i Contact0.

### 3.2.7. Negacija rezultata logičke operacije

Način rada (slika 3.10)



Slika 3.10. Negacija rezultata logičke operacije

Opis rada

Network 8:



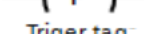
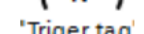
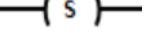
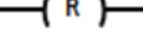
U prvom ljestvičastom dijagramu aktivira se „Coil13“ samo ako su neaktivni i Contact1 i Contact2. U drugom ljestvičastom dijagramu aktivira se Coil14 samo ako je neaktivan ili Contact4 i-ili Contact3.

### 3.3. Programiranje svitcima

Za izradu programa na raspolaganju je više simbola za izlaz. Osim što simboliziraju izlaz iz PLC-a koji može biti aktivan i neaktivan, moguće je ostvariti i posebne načine aktiviranja.

Simbol izlaza naziva se **svitak** (engl. *Coil*) što podsjeća na zavojnici releja ili kontaktora.

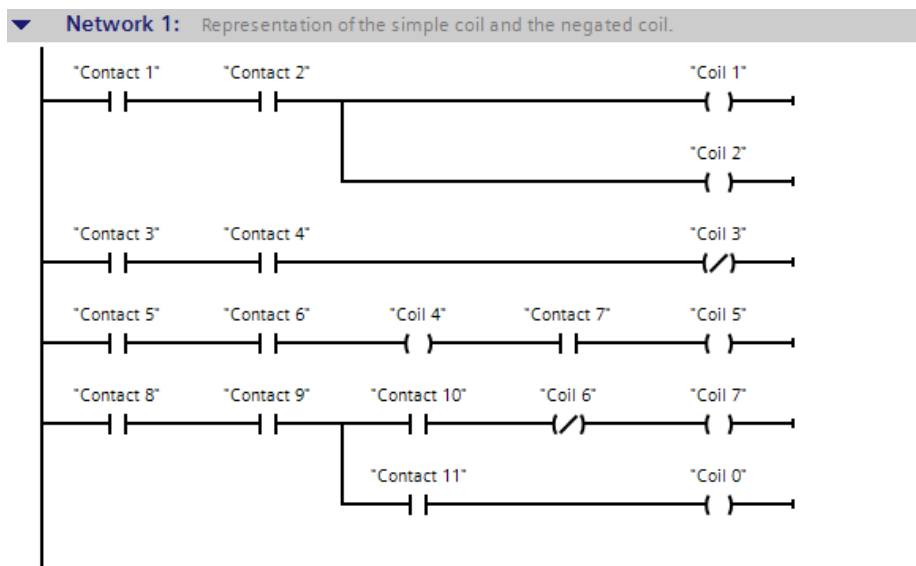
Na raspolaganju za izradu programa su sljedeći svitci: (slika 3.11)

Binarni tag 	Binarni tag 	Binarni tag  Triger tag	Binarni tag  'Triger tag'	Binarni tag 	Binarni tag 
Standardni svitak	Negacija stanja aktivacije	Puls na pozitivni brid	Puls na negativni brid	Set	Reset

Slika 3.11. Svitci

### 3.3.1. Standardni i negativni svitak

Način rada (slika 3.12):



Slika 3.12. Standardni i negativni svitak

Opis rada

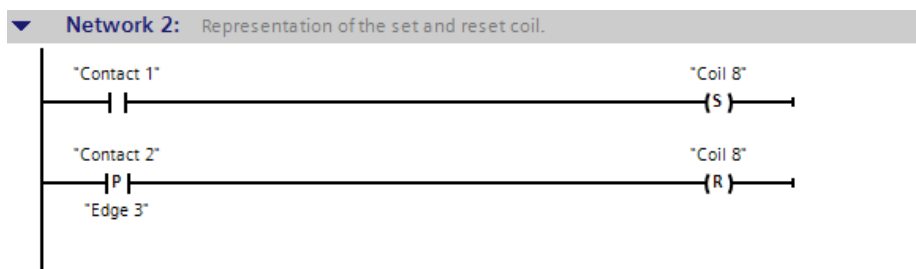
Network 1:

U prvom ljestvičastom dijagramu aktivira se Coil1 samo ako su aktivni Contact1 i Contact2.

U drugom ljestvičastom dijagramu aktivira se Coil3 samo ako je neaktivna serijska grana s Contact3 i Contact4, tj. bar jedan od njih nije aktivan.

### 3.3.2. Set i reset svitak

Način rada (lika 3.13):



Slika 3.13. Set i reset svitak

Opis rada

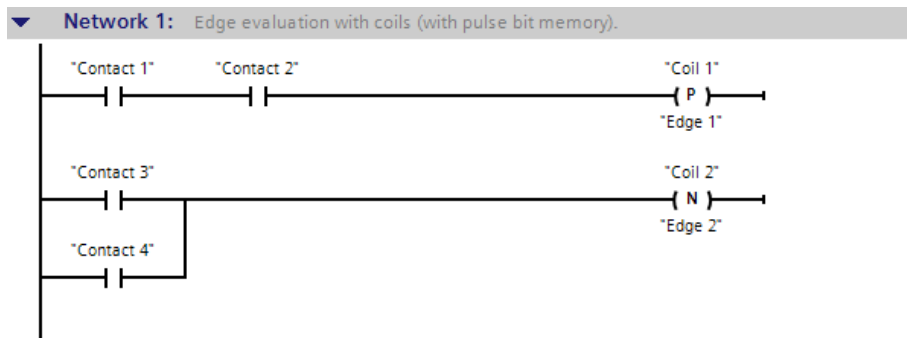
Network 2:

U prvom ljestvičastom dijagramu aktivira se Coil8 ako je aktivan Contact1.

U drugom ljestvičastom dijagramu deaktivira se Coil8 na pozitivan brid Contact2.

### 3.3.3. Pulsni izlaz

Način rada (slika 3.14):



Slika 3.14. Pulsni izlaz

Opis rada

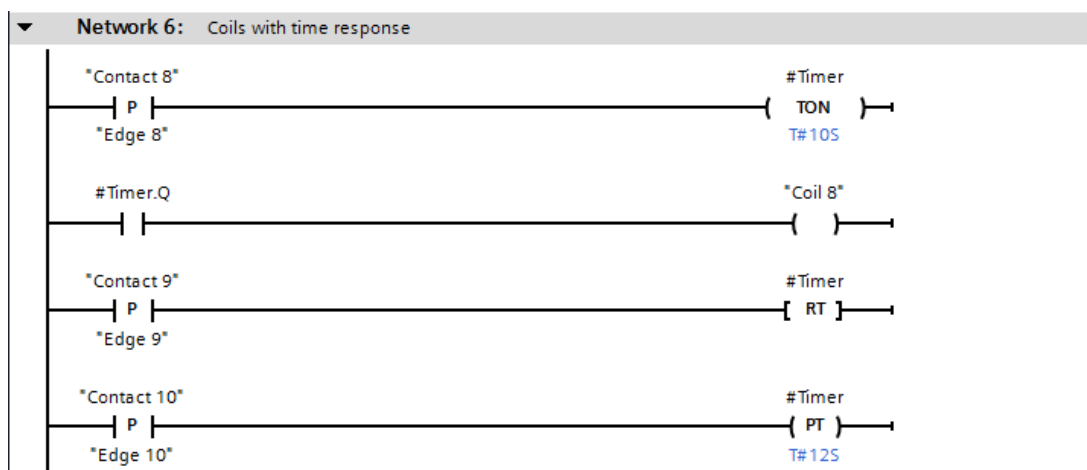
Network 1:

U prvom ljestvičastom dijagramu aktivira se Coil1 na pozitivan brid grane u kojoj se nalazi (aktivni Contact1 i Contact2).

U drugom ljestvičastom dijagramu aktivira se Coil2 na negativan brid grane u kojoj se nalazi (aktivan ili Contact3 i-ili Contact4).

### 3.3.4. IEC tajmer

Način rada (slika 3.15):



Slika 3.15. IEC tajmer

Tajmer (TON) uključuje se kada Contact8 propusti visoko stanje prema tajmeru. Tajmer nakon proteka 10s uključuje Coil8. Kad se tajmer aktivira, aktivira se i izlaz (RT) koji resetira postavljeni tajmer. U sljedećem koraku postavlja se nova vrijednost (PT) na 12s.

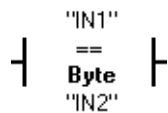
Na sličan način upravlja se i s druga tri tipa tajmera. Iako u ovom primjeru nisu korišteni, mogu se primijeniti (TP) pulsni tajmer, (TOF) tajmera po isključenju i (TONR) retentivni tajmer.

### 3.4. Funkcije – usporedba, premještanje i konverzija

#### Usporedba

Uspoređuju se dvije vrijednosti istog tipa podataka.

Tablica 3.3. Funkcija - usporedba

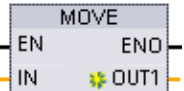
	== Usporedba je točna ako je IN1 jednak IN2.
	<> Usporedba je točna ako je IN1 različito od IN2.
	>= Usporedba je točna ako je IN1 veći ili jednak IN2.
	<= Usporedba je točna ako je IN1 manji ili jednak IN2.
	> Usporedba je točna ako je IN1 veći od IN2.
	< Usporedba je točna ako je IN1 manji od IN2.

Ako je usporedba točna, aktivira se izlazni kontakt.

#### Premještanje

Premještanje podataka s jedne adrese na drugu.

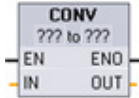


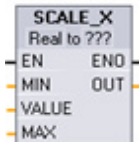
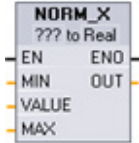
Tablica 3.4. Funkcija - premještanje

	S adrese koja se postavlja na ulaz bloka IN podatak se premješta na adresu odnosno tag naveden na OUT1.
---	---

#### Konverzija

Pretvaramo jedan tip podatka u drugi kako bismo dalje s njim mogli izvršavati određene operacije.

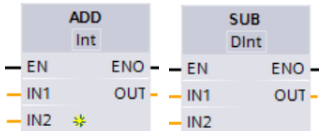
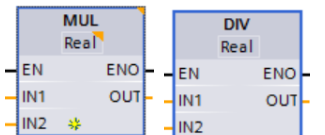
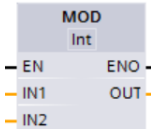
Tablica 3.5, Funkcija - konverzija

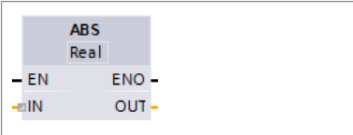
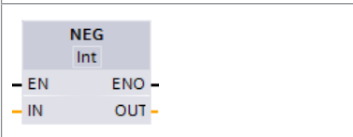
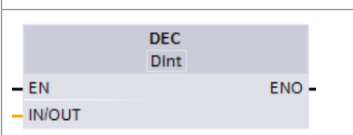
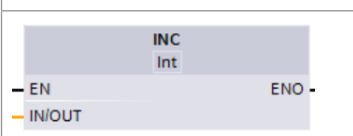
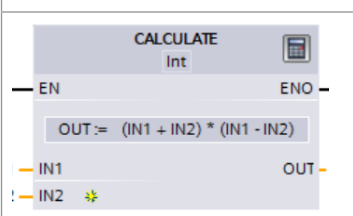
	<p>Pretvara jedan tip podatka u drugi kako bi se dalje s njim mogle izvršavati određene operacije.</p>
	<p>Pretvara decimalni broj (REAL) u cjelobrojni (DINT) i pritom ga zaokružuje na najbližu cjelobrojnu parnu vrijednost.  <math>\text{ROUND}(10.5)=10</math>  <math>\text{ROUND}(11.5)=12</math></p>
	<p>Pretvara decimalni broj (REAL) u cjelobrojni (DINT) i pritom ga zaokružuje na veću ili jednaku cjelobrojnu vrijednost.</p>
	<p>Skalira normaliziranu vrijednost VALUE pri čemu je <math>(0.0 \leq \text{VALUE} \leq 1.0)</math> u vrijednost unutar specificiranog opsega između MIN i MAX parametra:  <math>\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}</math></p>
	<p>Normalizira vrijednost VALUE unutar zadanog opsega između MIN i MAX vrijednosti:  <math>\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN})</math>,  pri čemu je <math>(0.0 \leq \text{OUT} \leq 1.0)</math></p>

### 3.5. Aritmetičke funkcije za numeričke vrijednosti

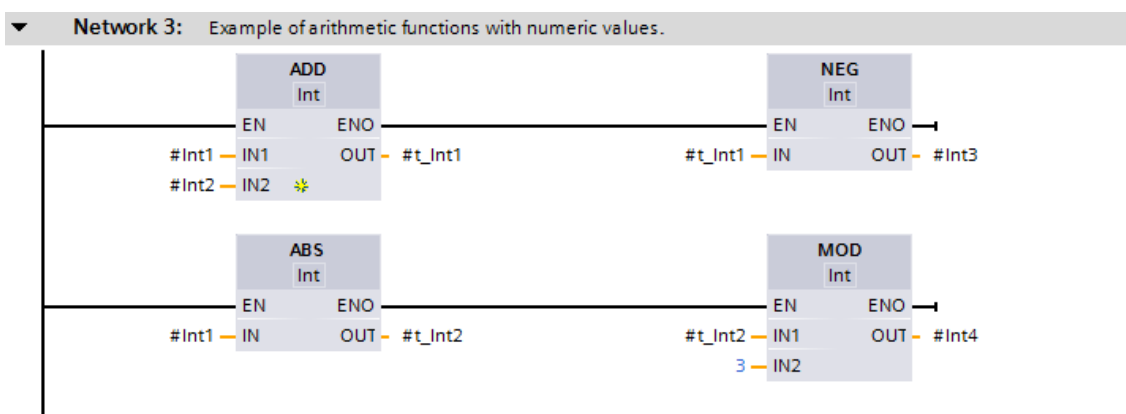
Sljedeće naredbe za aritmetičke funkcije dostupne su u programskom jeziku LAD:

Tablica 3.6, Aritmetičke funkcije u programskom jeziku LAD

	<p>zbrajanje (ADD)  <math>\text{OUT} = \text{IN1} + \text{IN2}</math></p> <p>oduzimanje (SUB)  <math>\text{OUT} = \text{IN1} - \text{IN2}</math></p>
	<p>množenje (MUL)  <math>\text{OUT} = \text{IN1} * \text{IN2}</math></p> <p>dijeljenje (DIV) dvije numeričkeih vrijednosti  <math>\text{OUT} = \text{IN1} / \text{IN2}</math></p>
	<p>dijeljenje s ostatkom kao rezultatom (MOD)  <math>\text{OUT} = \text{ostatak}(\text{IN1}/\text{IN2})</math></p>

	forma apsolutne vrijednosti (ABS), $OUT =  IN $
	negacija (NEG, množenje s -1), $OUT = -IN$
	smanjenje (DEC, smanjenje brojčane vrijednosti za 1) $OUT = IN - 1$
	povećanje (INC, povećanje numeričke vrijednosti za 1) $OUT = IN + 1$
	<ul style="list-style-type: none"> <li>- složena matematička funkcija</li> <li>- matematički izraz kreira se unutar te funkcije</li> <li>- u ovom primjeru</li> </ul> $OUT = (IN1 + IN2) * (IN1 - IN2)$

Primjer aritmetičkih funkcija s numeričkim vrijednostima (slika 3.16). Zbrajaju se dva broja formata INT, a međurezultat se sprema u privremeni tag #t\_Int1. Taj međurezultat se množi s -1 i upisuje u tag #Int3. Generira se apsolutna vrijednost taga #Int1; ta se vrijednost dijeli s 3, a ostatak dijeljenja zapisuje se u tag #Int4.

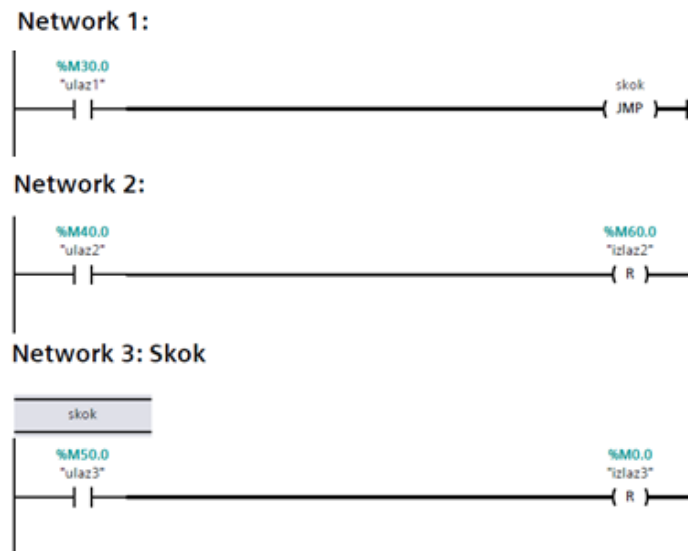


Slika 3.16. Primjer aritmetičkih funkcija u programskom jeziku LAD

## 3.6. Funkcije kontrole toka

### 3.6.1. Funkcija Jump

Način rada (slika 3.17):



Slika 3.17. Funkcija Jump

Opis rada:

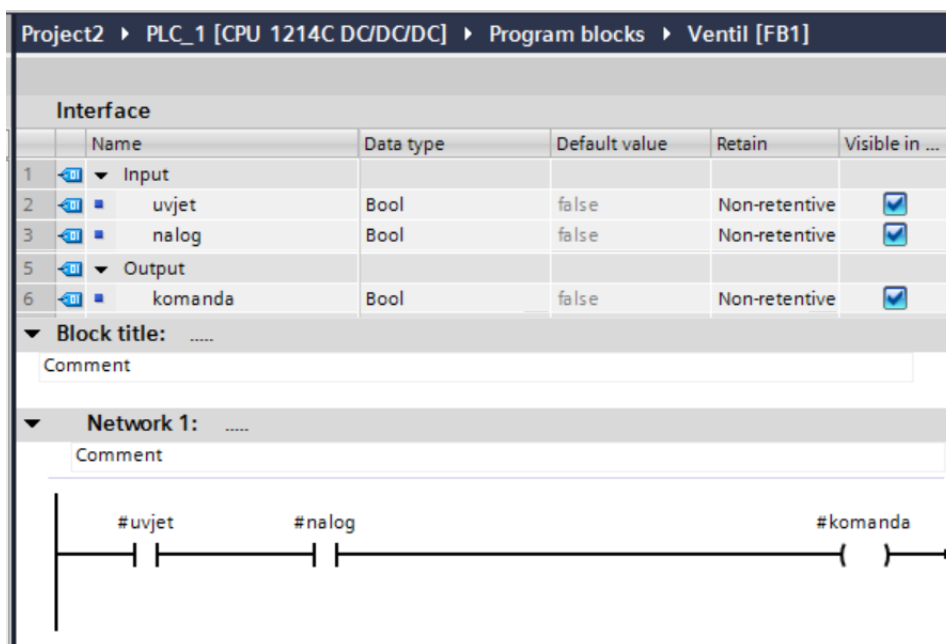
U mreži Network 3 nalazi se labela *skok*. U mreži Network 2 nalazi se funkcija (JMP). Ako je ispunjen prethodni uvjet, u slučaju da je „ulaz1“ aktivan, program preskače Network 2 i skače na labelu *skok*.

Napomena: Potrebno je biti vrlo oprezan u korištenju funkcije (JMP) kako program ne bi ušao u beskonačnu petlju iz koje ne može izaći.

### 3.6.2. Poziv funkcijskog bloka

Funkcijski blok, u ovom primjeru Ventil [FB1], ima dva ulazna parametra - uvjet i nalog. Ima jedan izlazni parametar naziva komanda.

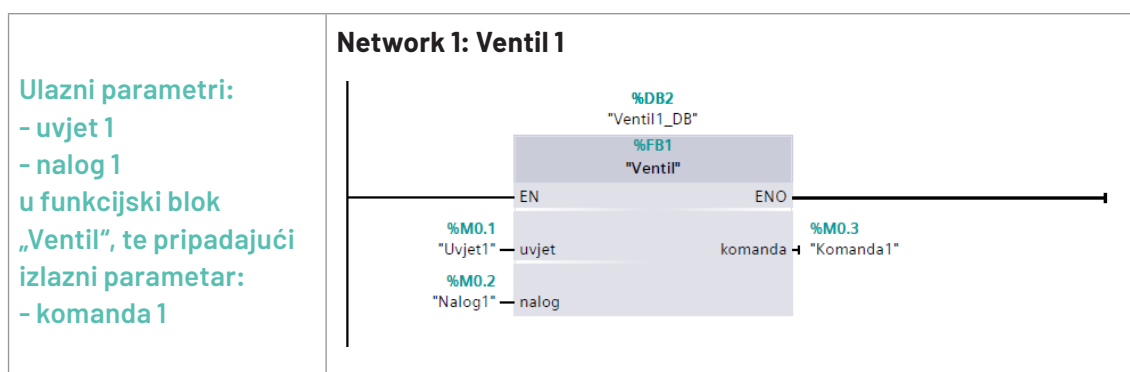
Njegova podatkovna struktura i programska struktura prikazani su na donjoj slici. Mreža Network 1, u ovom primjeru, predstavlja programsku strukturu funkcijskog bloka.



Slika 3.18. Funkcijski blok Ventil

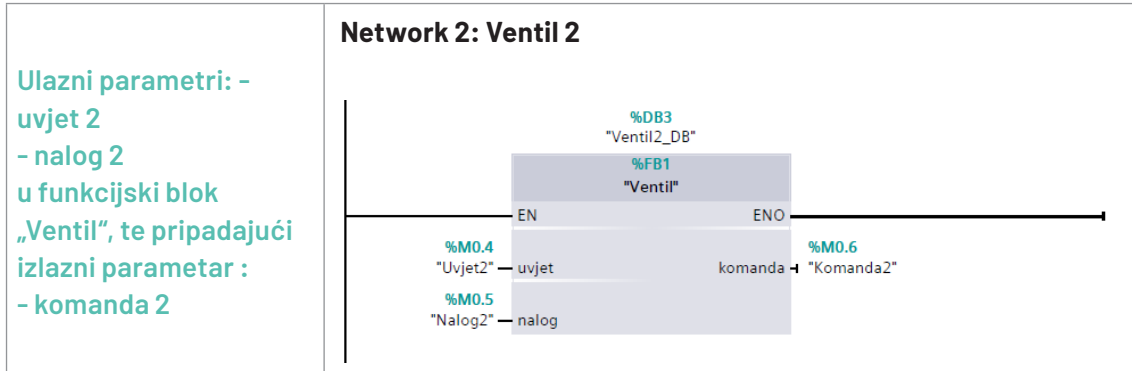
Pri pozivanju funkcijskog bloka može se definirati skup ulaznih i izlaznih parametara specifičan za mrežu u kojoj se nalazi poziv funkcijskog bloka. Pri svakom pozivu može se definirati različit skup parametara.

Funkcija iz koje se poziva taj funkcijski blok za ovaj primjer izgleda ovako:



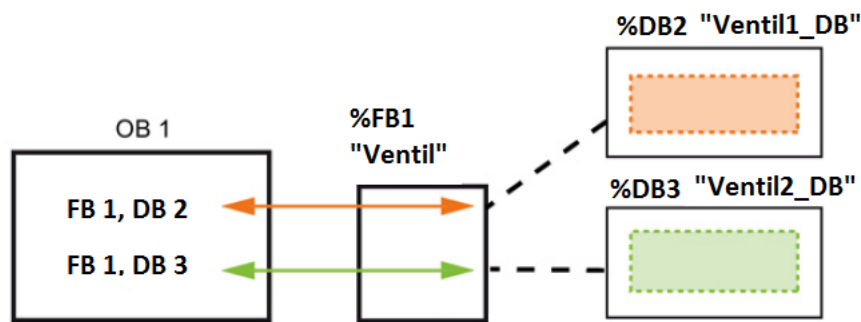
Slika 3.19. Poziv funkcijskog bloka Ventil za Ventil 1

Network 1 poziva funkcijski blok Ventil [FB1], uz ulazne parametre uvjet1 i nalog1 te izlazni parametar komanda1. Svi parametri se pritom smještaju u podatkovni blok %DB2 Ventil\_1\_DB.



Slika 3.20. Poziv funkcijskog bloka Ventil za Ventil 2

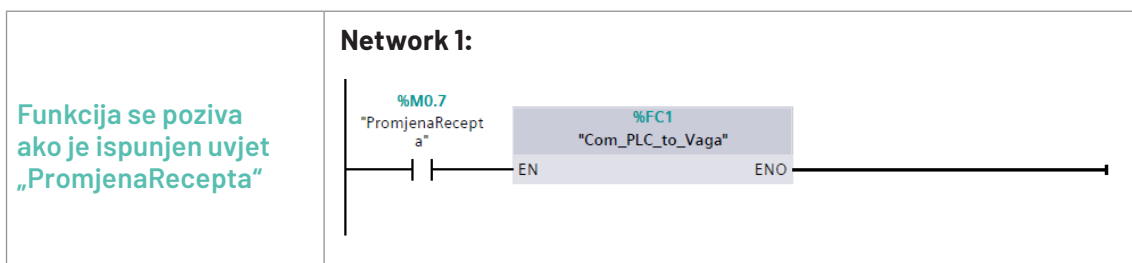
U mreži Network 2 funkcijski blok Ventil [FB1] poziva se uz ulazne parametre uvjet2, nalog2 te izlazni parametar komanda2. Svi parametri se pritom smještaju u podatkovni blok %DB3 Ventil\_2\_DB.



Slika 3.21. Poziv funkcijskog bloka

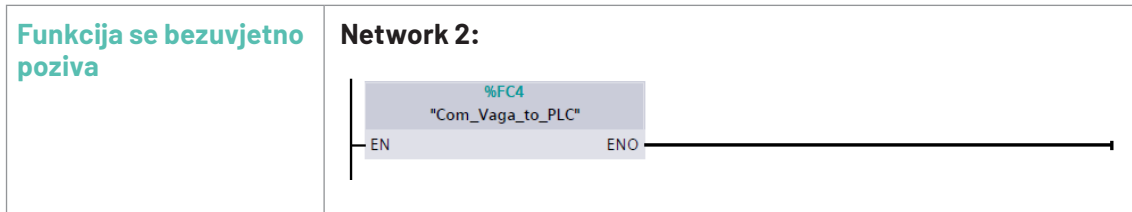
### 3.6.3. Poziv funkcije

Pozivanje funkcija može biti uvjetno ili bezuvjetno, kako je prikazano na sljedećim slikama (slika 3.22 i slika 3.23)



Slika 3.22. Poziv funkcije uvjetno

Ako uvjet ispred funkcije nije ispunjen, funkcija se neće pozivati.



Slika 3.23. Poziv funkcije bezuvjetno

Funkcija s prethodne slike poziva se svakim ciklusom programa.

## 4. Zadatci za vježbu

### 4.1. Zadatak 1: Temeljne funkcije ljestvičastog dijagrama

#### 4.1.1. Zadatak

Koristeći se tablicom ulaza i izlaza potrebno je napisati program u ljestvičastom dijagramu koji izvršava zadanu funkciju.

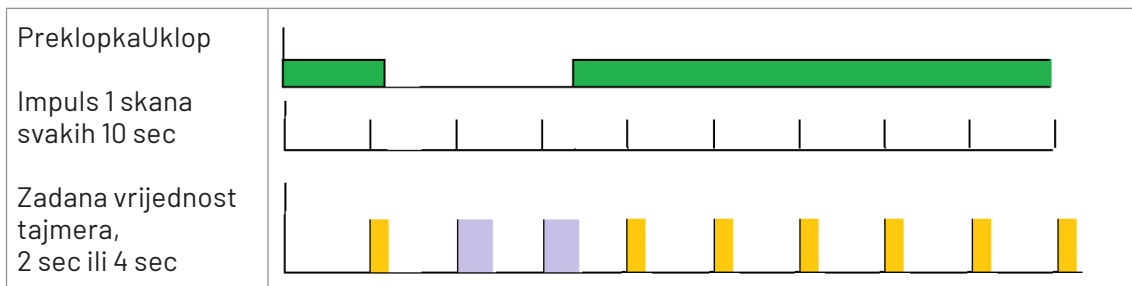
Tablica 4.1. Tablica ulaza i izlaza za Zadatak 1

Adresa	Simbol	Opis
I0.0	TipkaloStart	tipkalo za start
I0.1	TipkaloStop	tipkalo za stop
I0.2	PreklopkaUklop	preklopka uklopa
Q0.0	MotorM1	motor
PIW2	AnaUlaz	analogna vrijednost na ulazu
PQW2	Analzlaz	analogna vrijednost na izlazu

Zadana funkcija:

- A** - dok je aktivirano TipkaloStart, potrebno je uključiti izlaz MotorM1.
- B** - TipkaloStart uključuje izlaz MotorM1, a TipkaloStop ga isključuje.
- C** - PreklopkaUklop uključuje izlaz MotorM1, sa zadržskom od 10 sec.
- D** - TipkaloStart uvećava, a TipkaloStop umanjuje sadržaj brojača čija je zadana vrijednost 10. Izlaz MotorM1 uključuje se kad je postignuta zadana vrijednost brojača.
- E** - TipkaloStart odbrojava vrijednost u jedan brojač, a TipkaloStop u drugi brojač. Oba brojača imaju zadanu vrijednost 20. Svaki put kada su njihove trenutne vrijednosti izjednačene, uključuje se izlaz MotorM1.
- F** - ako je PreklopkaUklop aktivna, preslikaj vrijednost analognog ulaza PIW2 na izlaz PQW2, a ako je neaktivna, proslijedi vrijednost 0 na izlaz QW2.
- G** - retvori varijablu tipa INT u varijablu tipa REAL.
- H** - normiraj vrijednost analognog ulaza PIW2, koja je u rasponu od 0 do 27648, te je potom skaliraj u rasponu od 0 do 100 i postavi na analogni izlaz PQW2.

- I- postavi izlaz MotorM1 samo kada je bit br. 7 analognog ulaza jednak 1, neovisno o tome u kojem su statusu ostali bitovi analognog ulaza.
- J- svakih 10 sekundi postavljaj izlaz MotorM1 tako da trajanje impulsa ovisi o statusu PreklopkaUklopa - ako je aktivna, puls je trajanja 2 sec, a ako nije, puls je trajanja 4 sec. Način rada i vremenski dijagram prikazani su na slici 4.1.

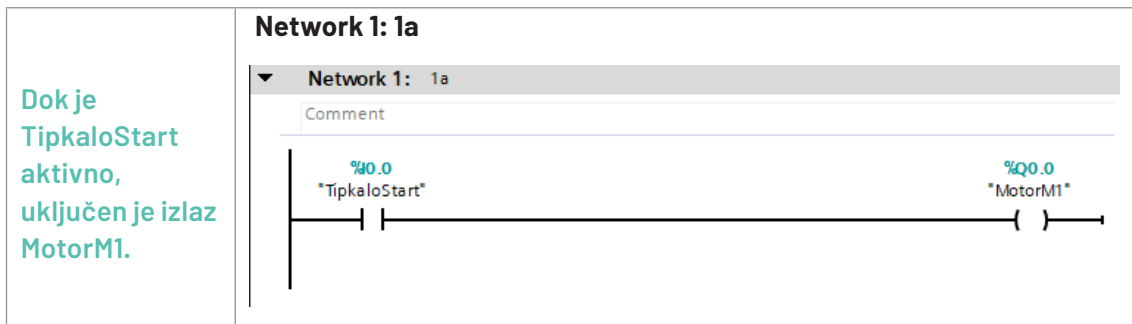


Slika 4.1. Zadatak 1j

#### 4.1.1. Rješenje

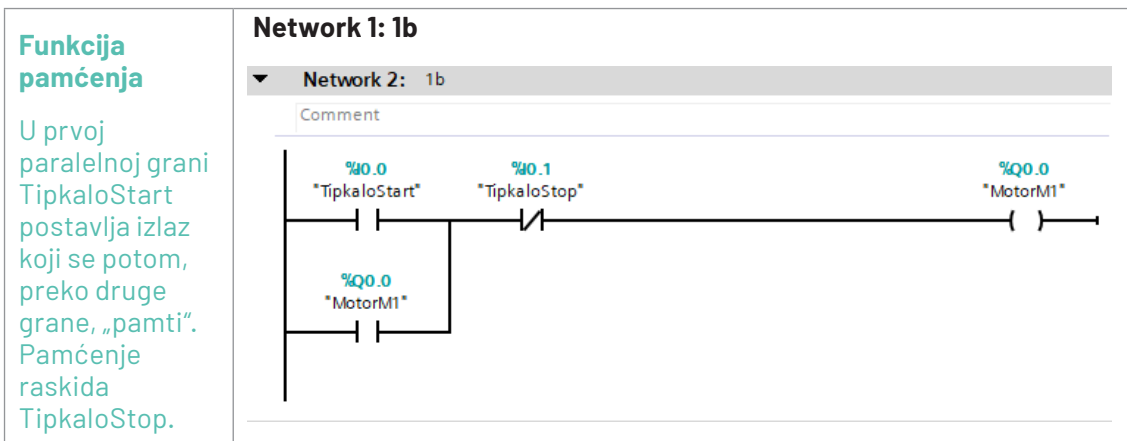
Ovdje je prikazano jedno od mogućih rješenja.

- A - TipkaloStart uključuje izlaz MotorM1

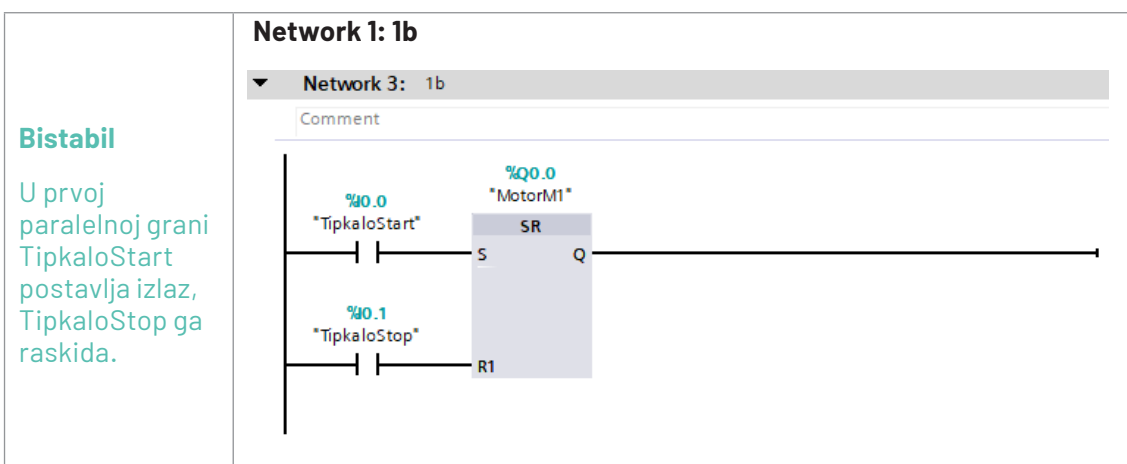


Slika 4.2 Rješenje 1a

- B - Prikazuju se dva rješenja zadatka: TipkaloStart uključuje izlaz MotorM1, a TipkaloStop ga isključuje.

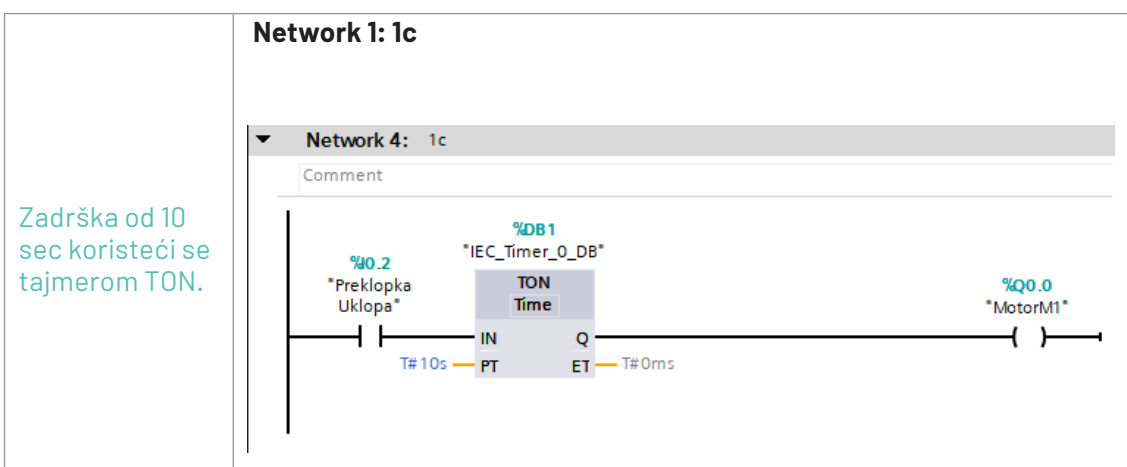


Slika 4.3. Rješenje 1b – funkcija pamćenja



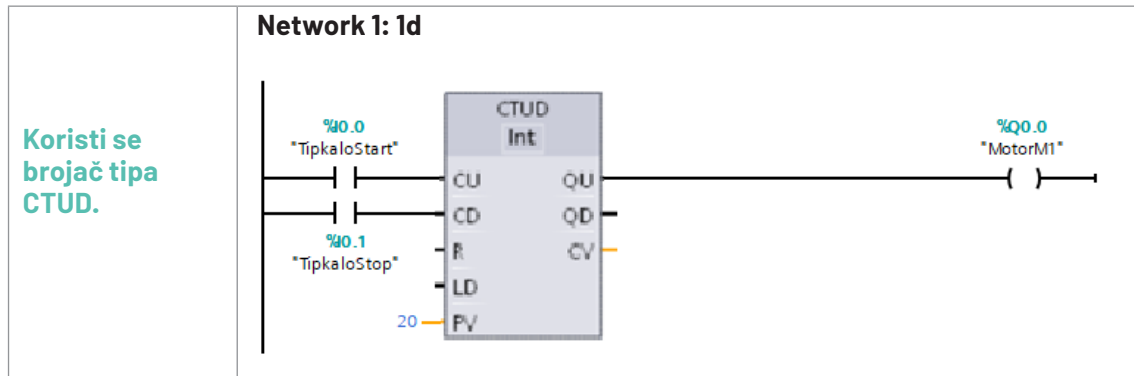
Slika 4.4. Rješenje 1b – bistabil

**C** - PreklopkaUklop uključuje izlaz MotorM1 sa zadržkom od 10 sec.



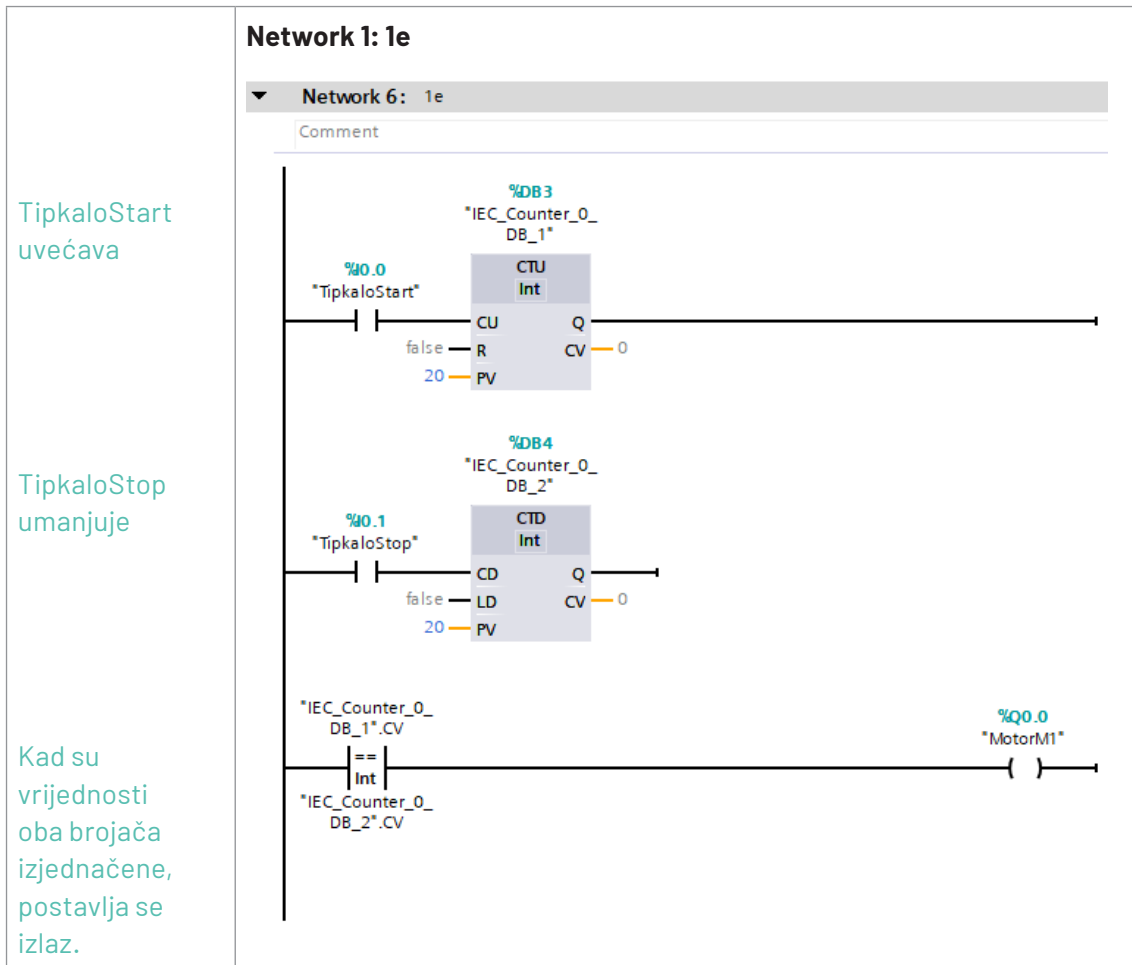
Slika 4.5. Rješenje 1c

- D** - TipkaloStart uvećava, a TipkaloStop umanjuje vrijednost brojača čija je zadana vrijednost 10. Izlaz MotorM1 uključuje se kad je postignuta zadana vrijednost brojača.



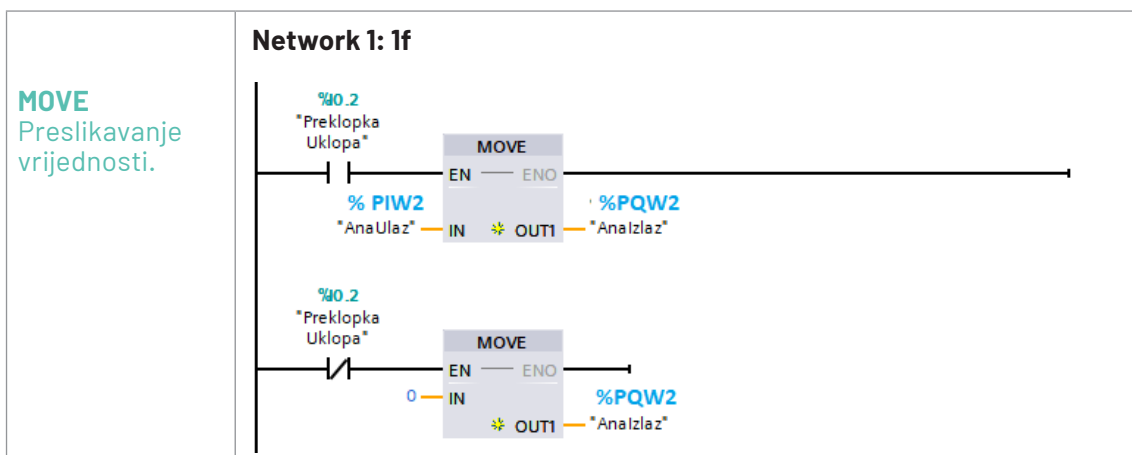
Slika 4.6. Rješenje 1d

- E** - TipkaloStart uvećava vrijednost u jedan brojač, a TipkaloStop u drugi. Oba brojača imaju zadanu vrijednost 20. Svaki put kada su njihove trenutne vrijednosti izjednačene, uključuje se izlaz MotorM1.



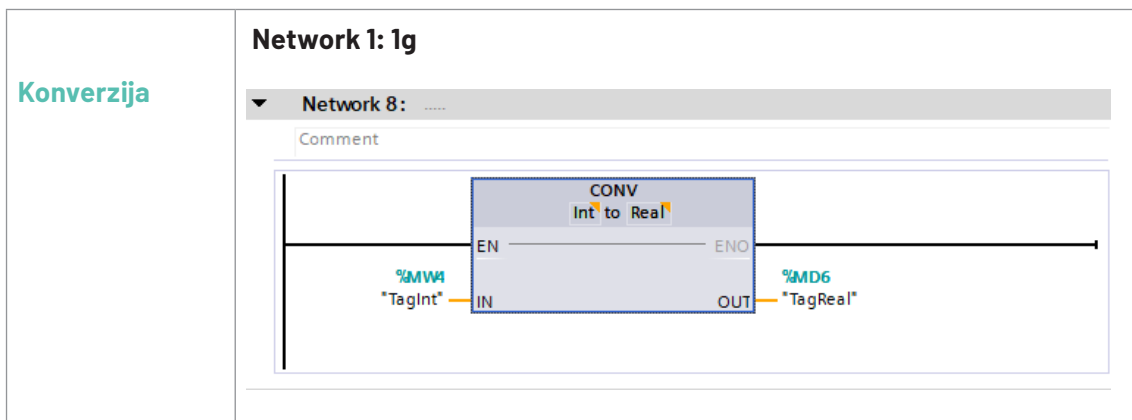
Slika 4.7. Rješenje 1e

F- ako je PreklopkaUklop aktivna, preslikaj vrijednost analognog ulaza PIW2 na izlaz PQW2, a ako je neaktivna, proslijedi vrijednost 0 na izlaz QW2.



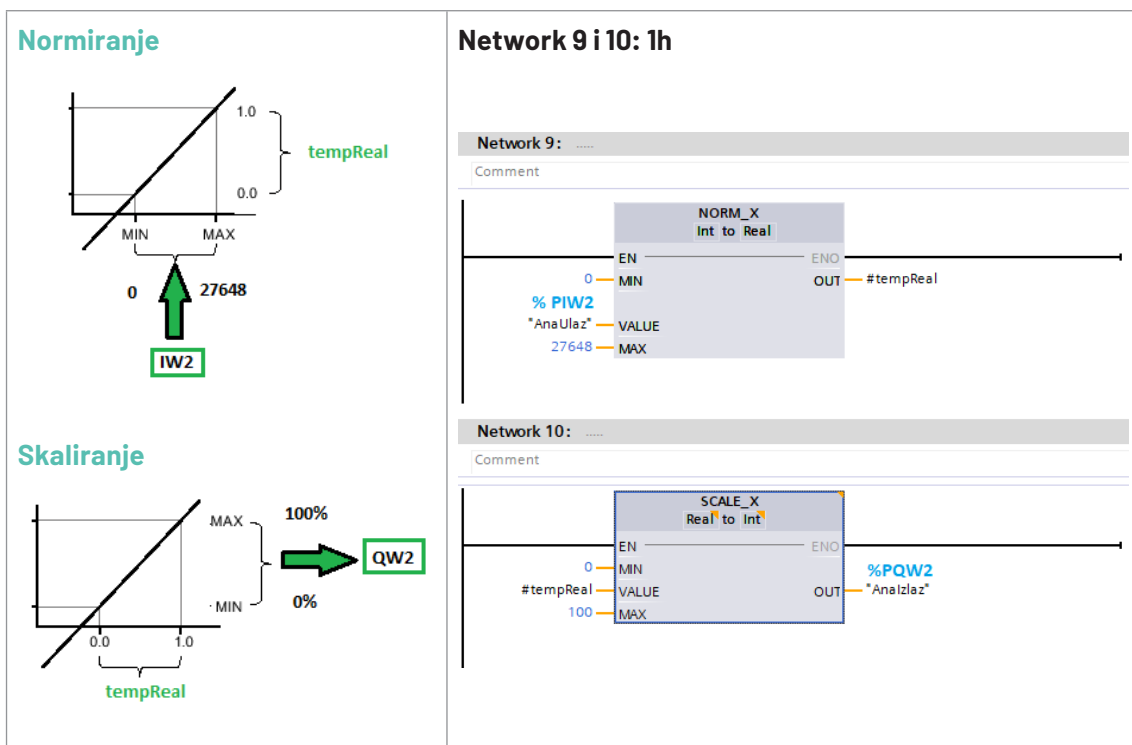
Slika 4.8. Rješenje 1f

G - pretvori varijablu tipa INT u varijablu tipa REAL.



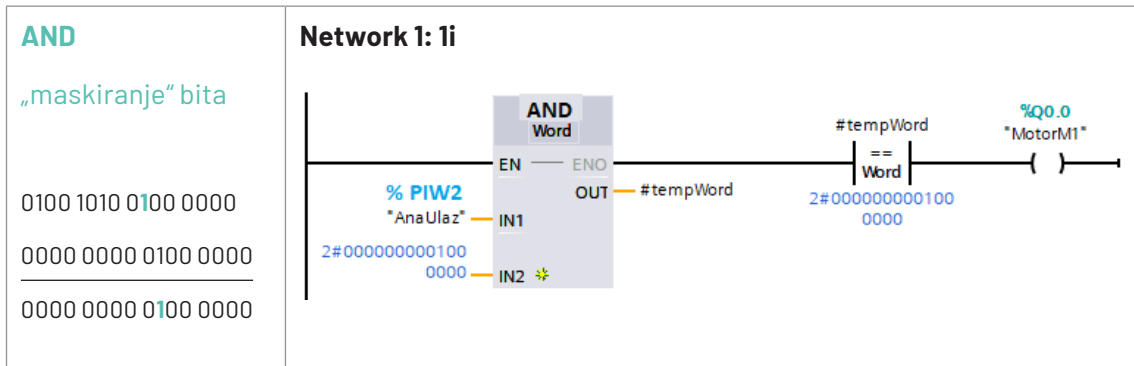
Slika 4.9. Rješenje 1g

H - normiraj vrijednost analognog ulaza PIW2, koja je u rasponu od 0 do 27648 te je potom skaliraj u rasponu od 0 do 100 i postavi na analogni izlaz PQW2.



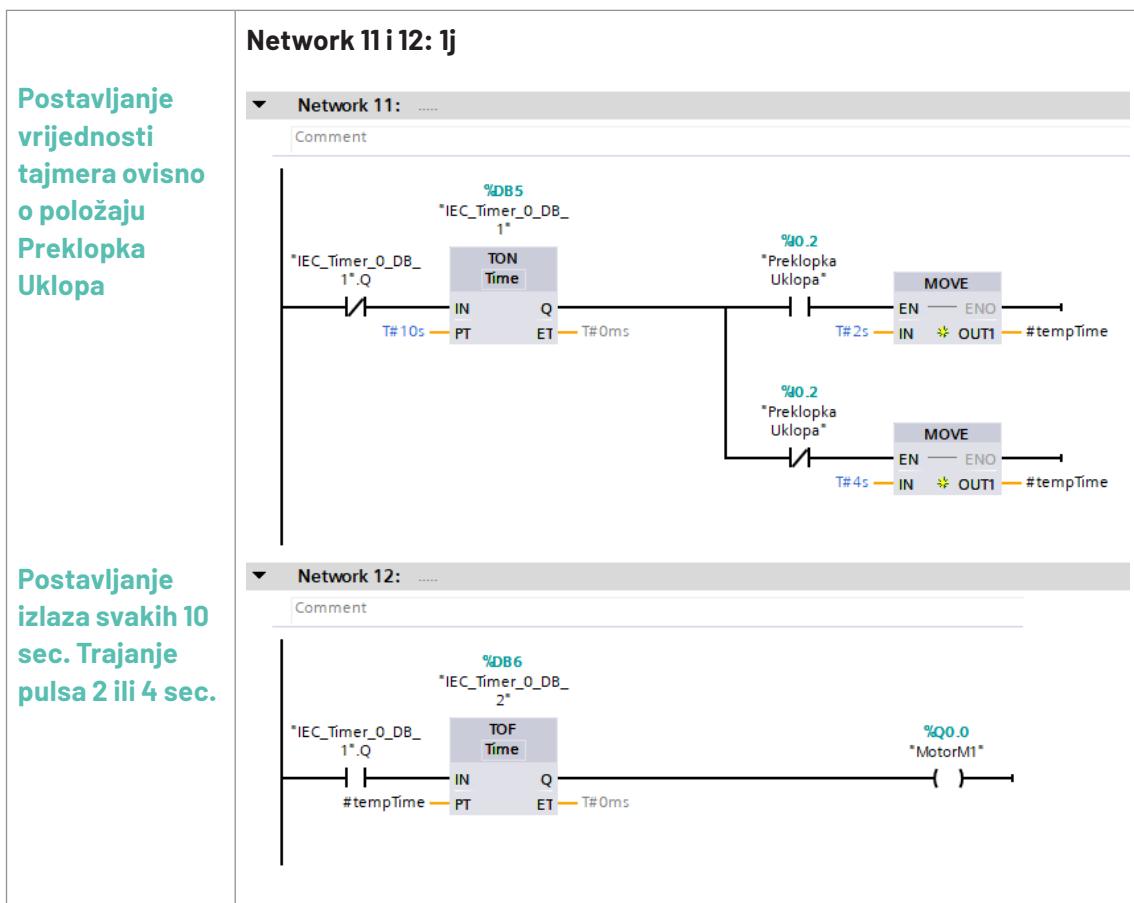
Slika 4.10. Rješenje 1h

- I- postavi izlaz MotorM1 samo kada je bit br. 7 analognog ulaza jednak 1, neovisno o tome u kojem su statusu ostali bitovi analognog ulaza.



Slika 4.11. Rješenje I1

- J- svakih 10 sec postavljaj izlaz MotorM1 tako da trajanje impulsa ovisi o statusu PreklopkaUklopa - ako je aktivna, puls je trajanja 2 sec, a ako nije, puls je trajanja 4 sec.



Slika 4.12. Rješenje Ij

## 4.2. Zadatak 2: Trčeće svjetlo

### 4.2.1. Zadatak

Potrebno je napisati program za trčeće svjetlo. Program uključuje i isključuje izlaze u slijedu, pri čemu je svaki izlaz u stanju logičke 1 jednak vremenski interval (period).

Tajmerom je određeno trajanje cijele sekvence.

Vremenskih intervala ima 6 - u prvom se uključuje lampica I1 koja se nalazi na adresi Q0.0, u drugom intervalu uključuje se lampica I2 i tako redom.

Nakon što se uključi zadnji izlaz, sekvenca se ponavlja.

Pritiskom na tipkalo potrebno je promijeniti smjer uključivanja izlaza.

Tablica 4.2. Tablica ulaza i izlaza za Zadatak 2

Adresa	Simbol	Opis
I0.0	start_stop	Start/STOP
I0.1	smjer	smjer lijevo/desno
Q0.0	I1	lampica
Q0.1	I2	lampica
Q0.2	I3	lampica
Q0.3	I4	lampica
Q0.4	I5	lampica
Q0.5	I6	lampica

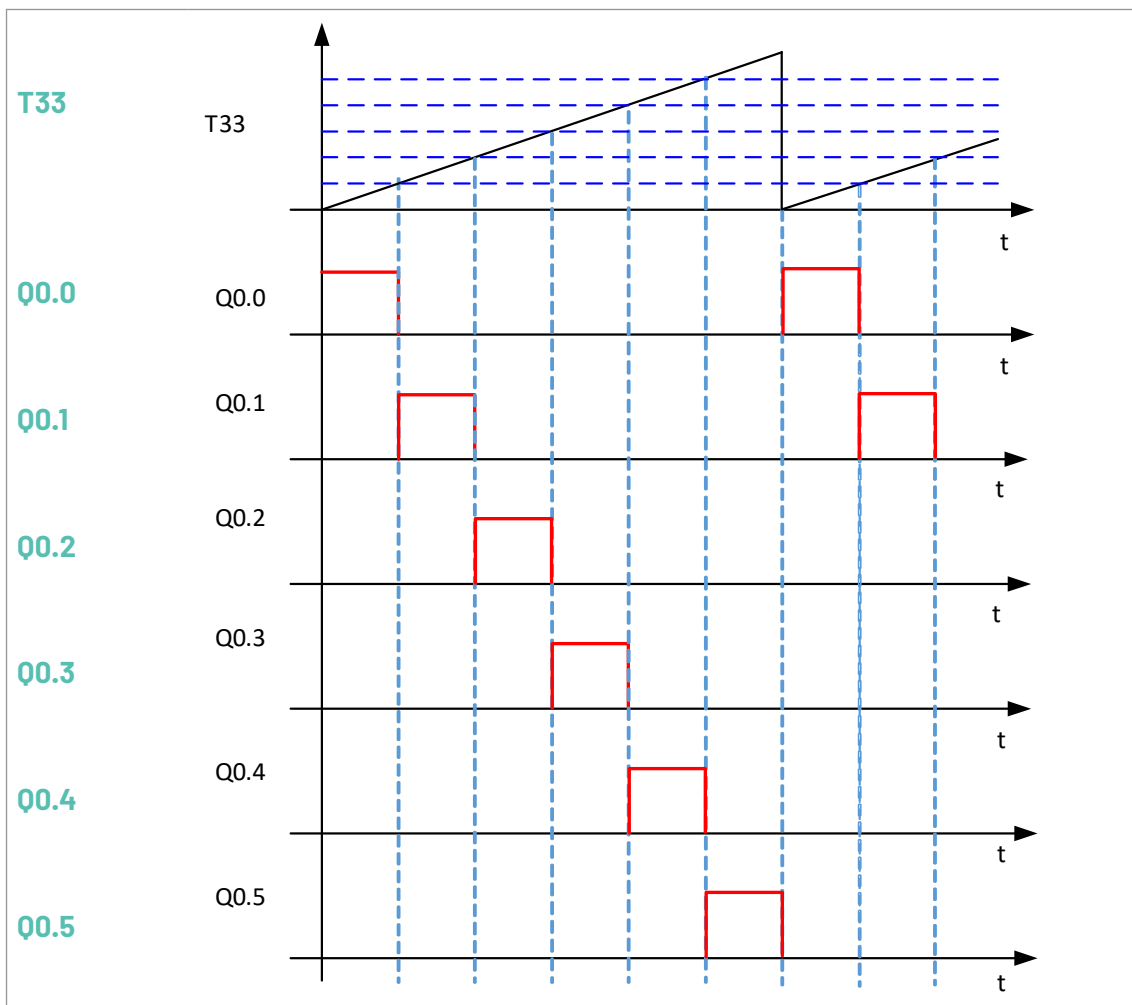
Napisati temeljne programske module koji generiraju signal T33 primjenom sljedećih tajmera:

Tajmer po uključanju, *ON delay timer*

Tajmer po isključenju, *OFF delay timer*

Akumulirajući tajmer, *Retentive timer*

Vremenski dijagram uključivanja i isključivanja lampica dan je u grafičkom obliku:



Slika 4.13. Vremenski dijagram uključivanja izlaza za Zadatak 2

#### 4.2.2. Rješenje

Ovdje je prikazano jedno od mogućih rješenja.

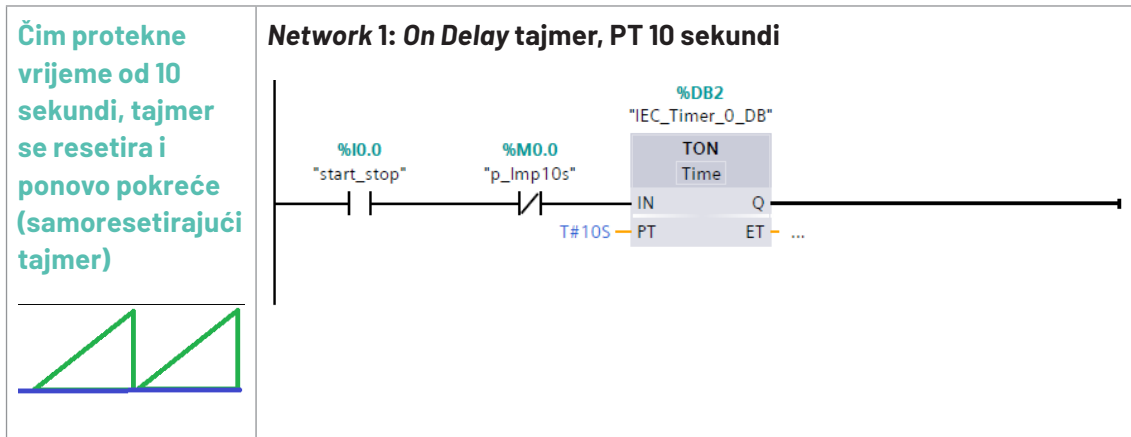
Zadatak zahtijeva da se programski modul riješi i s:

- A- tajmerom po uključanju
- B- tajmerom po isključanju
- C- akumulirajućim tajmerom

U svim rješenjima razlikuju se prve dvije mreže. Prve dvije mreže predstavljaju generator impulsa realiziran pomoću jednog od navedenih tajmera. Mreže 3 i 4 služe za postavljanje izlaza i identične su za sva tri slučaja.

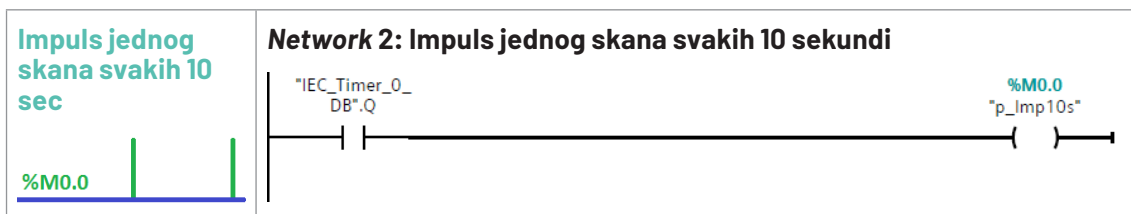
### Generiranje impulsa s tajmerom po uključenju - Network 1 i 2 (A)

Sljedeće dvije slike (slika 4.2 i slika 4.3) prikazuju prve dvije mreže za Rješenje A – rješenje s tajmerom po uključenju.



Slika 4.14. Rješenje A – Network 1 za Zadatak 2

Po uključenju tajmer svoj izlaz Q postavlja u vrijednost 1 kada odbroji zadano vrijeme. U idućem skanu tajmer ponovo počinje brojanje, a njegov izlaz Q postavlja se u vrijednost „0“. Na isti način postavlja se i %M0.0.

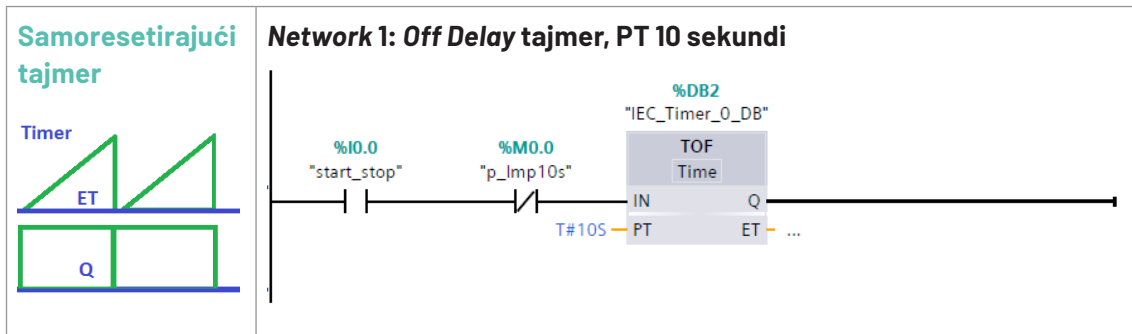


Slika 4.15. Rješenje A – Network 2 za Zadatak 2

%M0.0 postavlja se u vrijednost 1 svakih 10 sekundi u trajanju od 1 skana.

### Generiranje impulsa s tajmerom po isključenju, Network 1 i 2 (B)

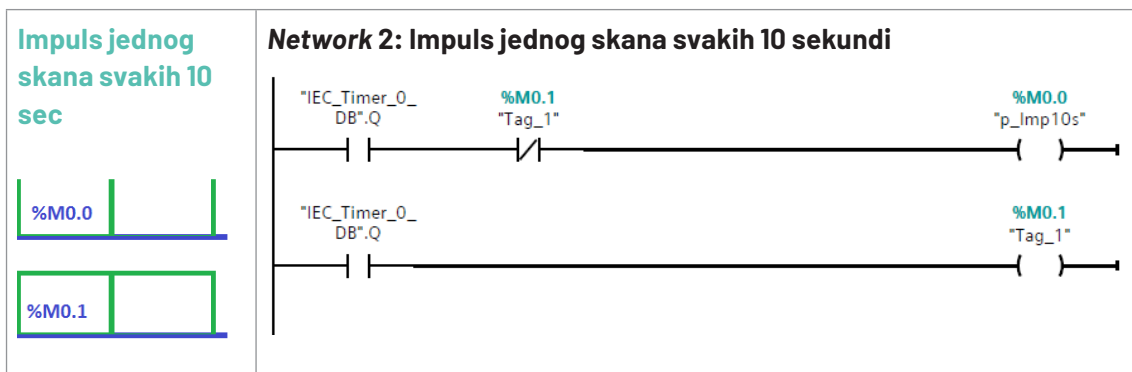
Sljedeće dvije slike (slika 4.4 i slika 4.5) prikazuju prve dvije mreže za Rješenje B – rješenje s tajmerom po isključenju.



Slika 4.16. Rješenje B – Network 1 za Zadatak 2

Tajmer po isključenju svoj izlaz Q postavlja u vrijednost 1 čim započne brojanje. Kada odbroji zadano vrijeme, postavlja ga u vrijednost 0 u trajanju jednog skana.

Na isti način, postavlja se i %M0.1.

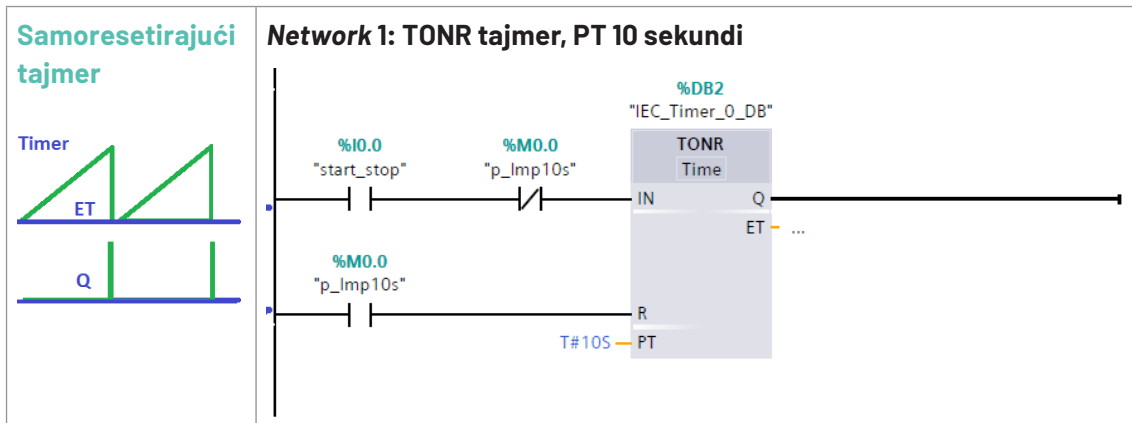


Slika 4.17. Rješenje B – Network 2 za Zadatak 2

%M0.0 je signal komplementaran signalu %M0.1, on je vrijednost 0 svakih 10 sekundi u trajanju od 1 skana.

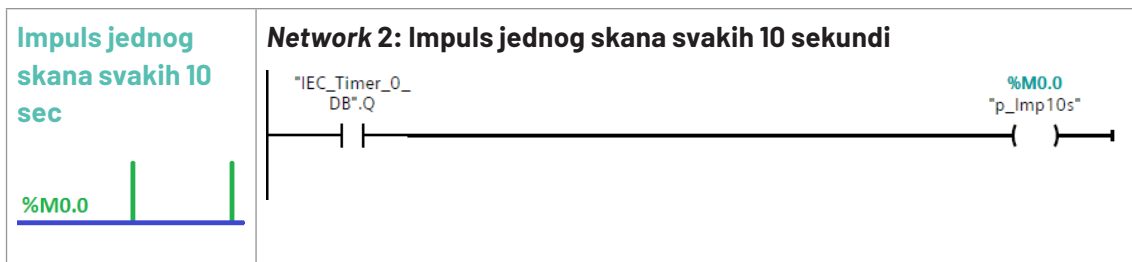
### Generiranje impulsa s akumulirajućim tajmerom, Network 1 i 2 (C)

Sljedeće dvije slike (slika 4.6 i slika 4.7) prikazuju prve dvije mreže za Rješenje B – rješenje s tajmerom po isključenju.



Slika 4.18. Rješenje C - Network 1 za Zadatak 2

Akumulirajući tajmer svoj izlaz Q postavlja u vrijednost 1 kada odbroji zadano vrijeme. Taj izlaz bio bi aktivan do daljnjega, sve dok ima uvjete na ulazu u tajmer pa ga za ovu priliku treba resetirati preko njegovog ulaza za reset, kako bi ispočetka započeo brojanje.



Slika 4.19. Rješenje C - Network 2 za Zadatak 2

Čim akumulirajući tajmer odbroji zadano vrijeme i postavi svoj izlaz Q u vrijednost 1, u istom skanu postavlja se i %M0.0. U idućem skanu %M0.0 resetira tajmer, izlaz Q postavlja se u vrijednost 0, a time i %M0.0.

### Postavljanje izlaza. Primjenjuje se identično za sva tri rješenja (A, B, C)

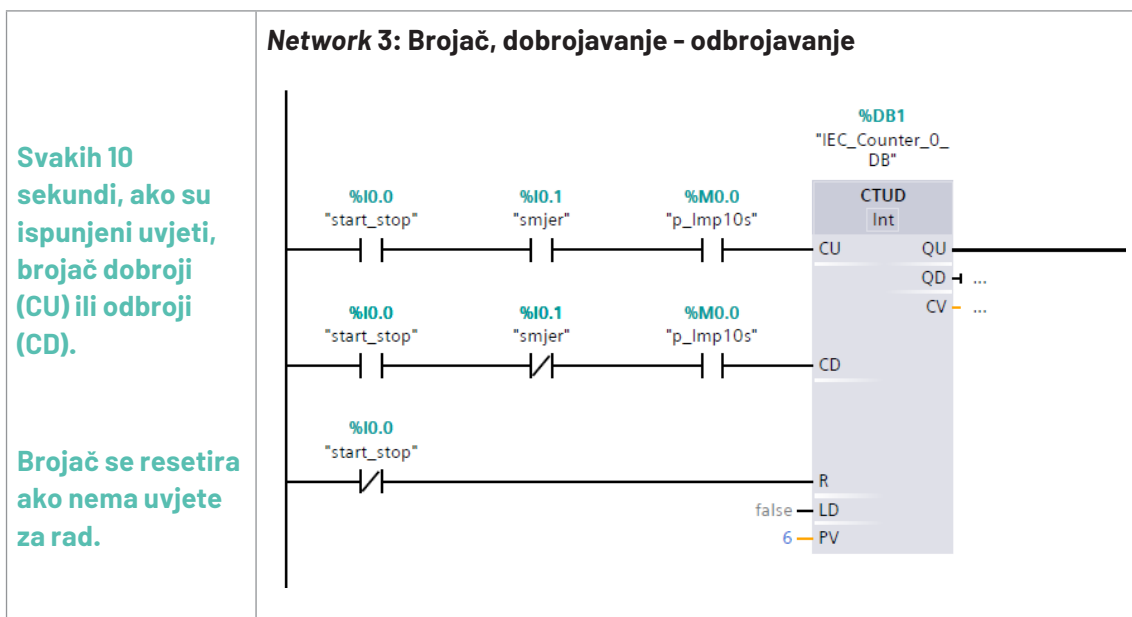
Impuls u trajanju od jednog skana generira se u mrežama *Network 1* i *Network 2* pomoću jednog od tajmera:

A- tajmera po uključenju

B- tajmera po isključenju

C- akumulirajućeg tajmera

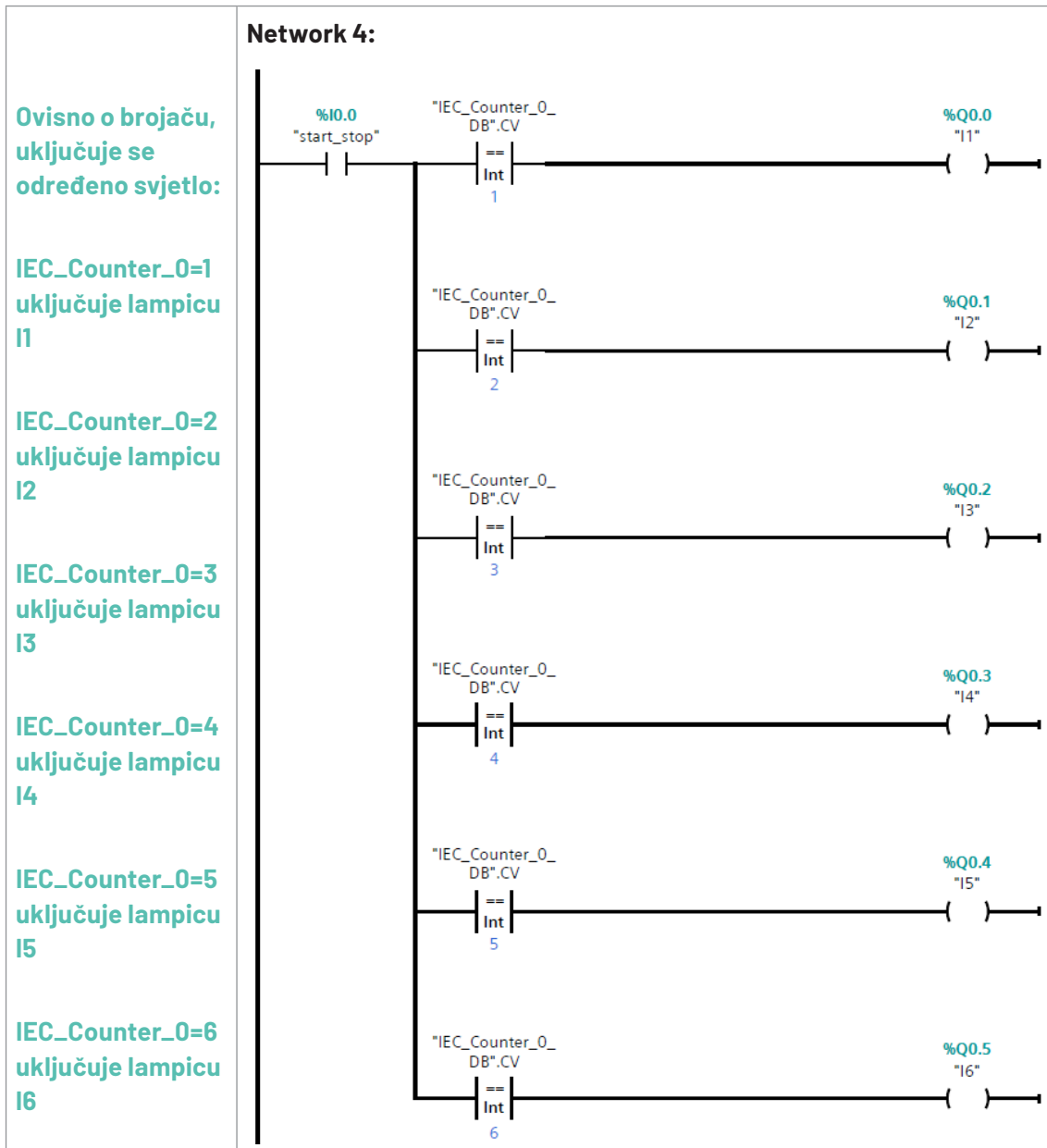
Kontakt `start_stop`, ako je u vrijednosti 1, omogućava ili zaustavlja brojanje impulsa koje je generirao tajmer.



Slika 4.20. Rješenje A,B,C – aktiviranje brojača za Zadatak 2

Ovisno o kontaktu „smjer“, ako kontakt ima vrijednosti 1, aktivira se brojanje naviše. Ako kontakt ima vrijednosti 0, aktivira se brojanje naniže i to u ritmu impulsa svakih 10 sekundi.

Ako je uvjet `start_stop` aktivan, ovisno o sadržaju brojača, postavlja se odgovarajući izlaz u aktivno stanje. Izlaz je aktivan u trajanju jednog perioda tajmera. Postavljanje izlaza, odnosno uključivanje na izlaze povezanih lampica, realizira se u mreži *Network 4*.



Slika 4.21. Rješenje A,B,C - postavljanje izlaza za Zadatak 2

### 4.3. Zadatak 3: Semafor

#### 4.3.1. Zadatak

Napisati program za upravljanje prometnom signalizacijom.

Prometna signalizacija predstavlja sustav semafora za upravljanje prometom na raskrižju. Program treba u automatskoj sekvenci uključivati svjetla na semaforu u oba smjera prema priloženom vremenskom dijagramu.

Postaviti vremenski ciklus na 10 sekundi ( $t_{ON}=10s$ ).

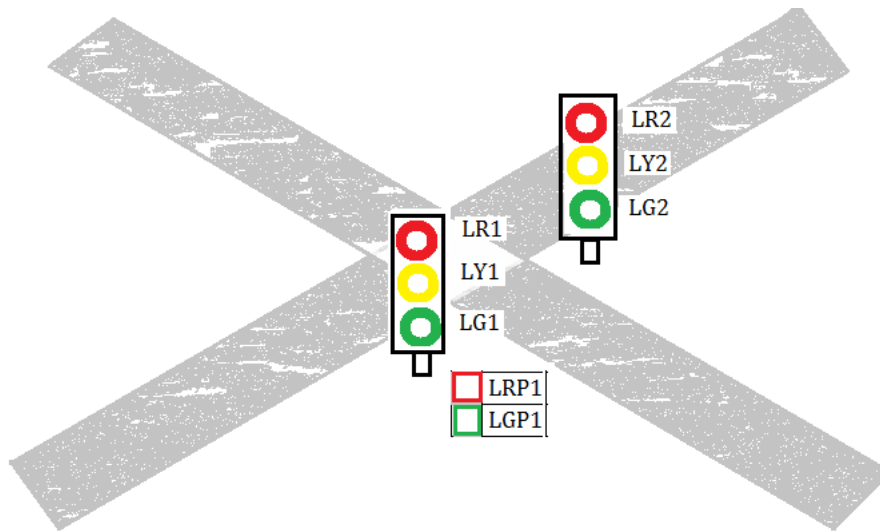
Tablica 4.3. Tablica ulaza i izlaza za Zadatak 3

Adresa	Simbol	Opis
I0.0	S1	Start/STOP
I0.1	S2	standardan rad/žuto treptajuće svjetlo
Q0.0	LR1	crveno svjetlo smjer 1.
Q0.1	LY1	žuto svjetlo smjer 1.
Q0.2	LG1	zeleno svjetlo smjer 1.
Q0.3	LR2	crveno svjetlo smjer 2.
Q0.4	LY2	žuto svjetlo smjer 2.
Q0.5	LG2	zeleno svjetlo smjer 2.
Q0.6	LRP1	zeleno svjetlo pješaci smjer 1.
Q0.7	LGP1	crveno svjetlo pješaci smjer 1.

#### S1=0

(semafori ne rade)

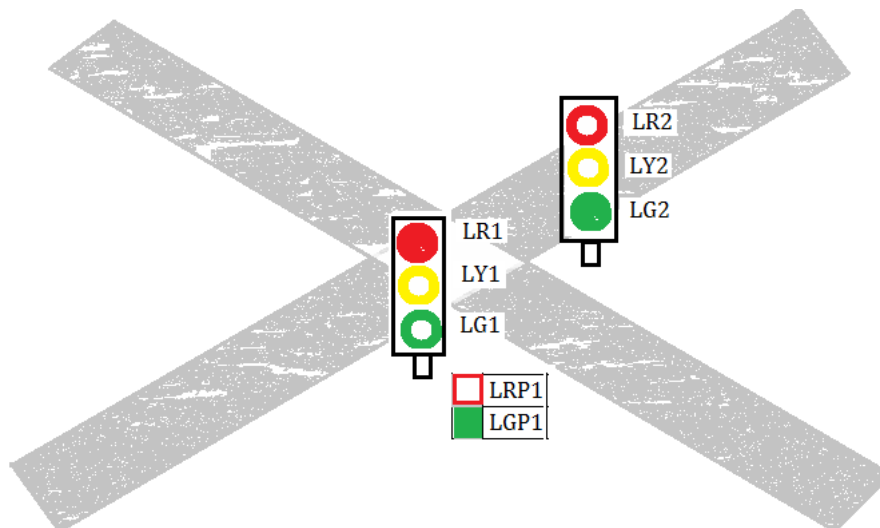
Semafori ne rade, ni cestovni ni pješачki.

Slika 4.22. Rješenje  $S1=0$  za Zadatak 3 **$S1=1$  i  $S2=1$** 

(standardni rad semafora)

Semafori rade u skladu sa zadanim vremenskim dijagramima.

Npr. za period 1, tj. za područje 1: od 0-4 sekunde

Slika 4.23. Rješenje  $S1=1$  i  $S2=1$  za Zadatak 3

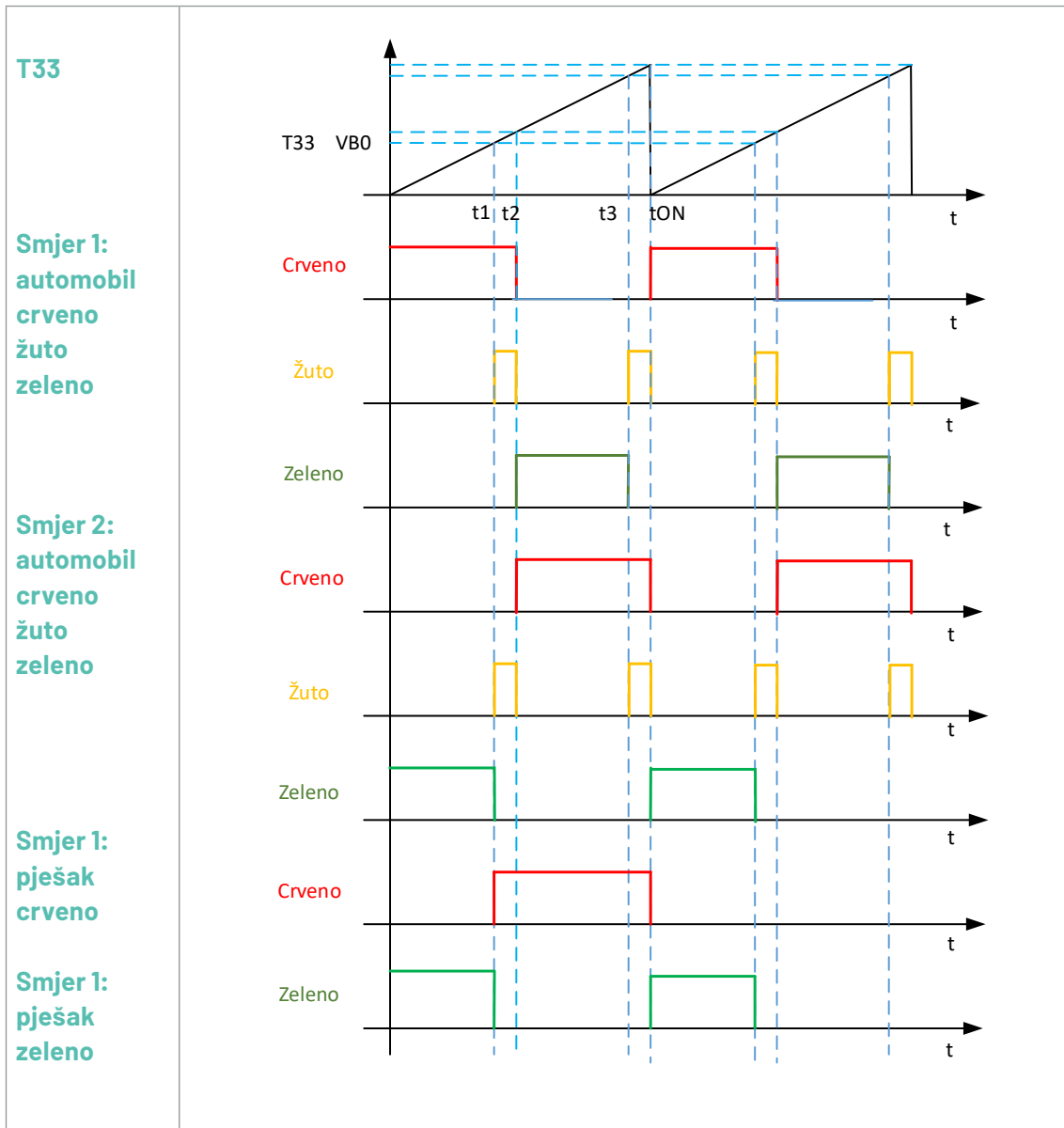
Vremenski dijagram uključenja i isključenja dan je na dva načina:

- tablično (tablica 4.3)

- grafički (slika 4.12)

Tablica 4.4. Vremenski dijagram uključenja i isključenja svjetala – tablično za Zadatak 3

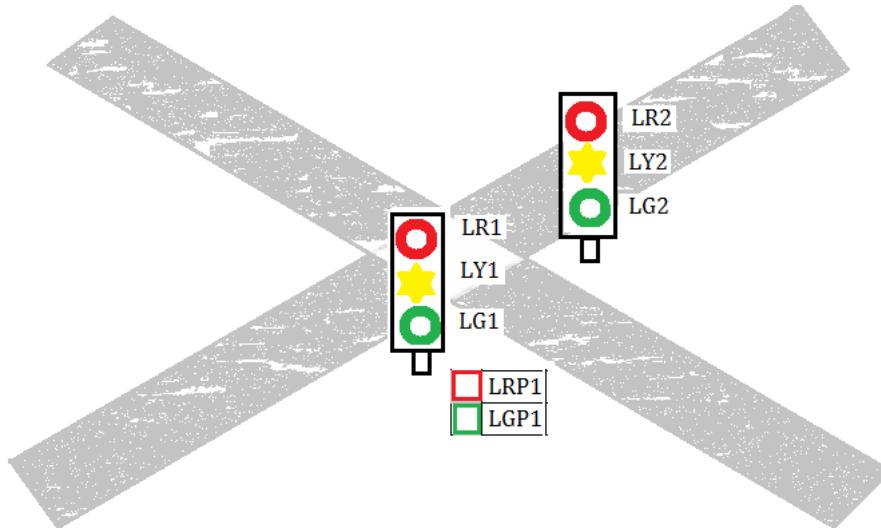
Period	1	2	3	4
LR1 automobil, smjer1, crveno				
LY1 automobil, smjer1, žuto				
LG1 automobil, smjer1, zeleno				
LR2 automobil, smjer2, crveno				
LY2 automobil, smjer2, žuto				
LG2 automobil, smjer2, zeleno				
LRP1 pješak, smjer1, crveno				
LGP1 pješak, smjer1, zeleno				



Slika 4.24. Vremenski dijagram uključenja i isključenja svjetala – grafički za Zadatak 3

**S1=1 i S2=0**

(žuto treptajuće svjetlo)

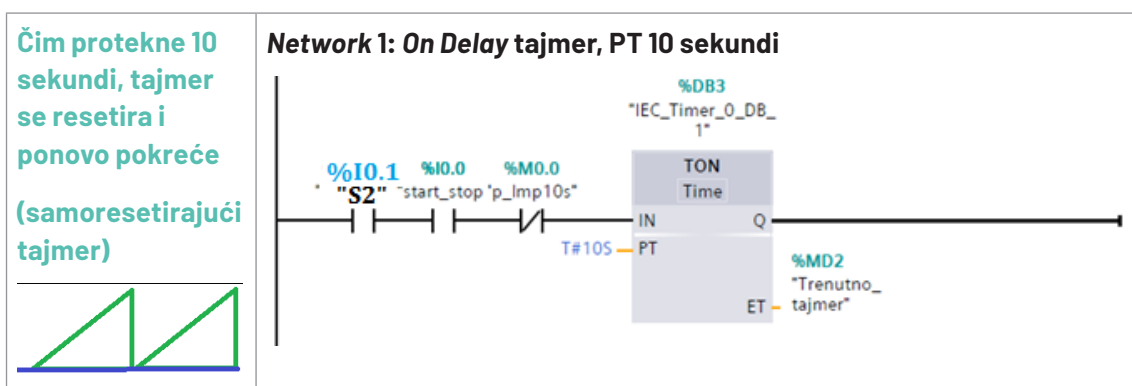


Slika 4.25. Rješenje S1=1 i S2=0 za Zadatak 3

### 4.3.2. Rješenje

Prvi korak u programu je realizirati generator pilastog valnog oblika. Pilasti valni oblik je iznos protoka vremena tajmera koji mijenja svoju vrijednost naviše svaku milisekundu. Kada tajmer izbroji (PT) milisekundi, resetira se na početnu vrijednost.

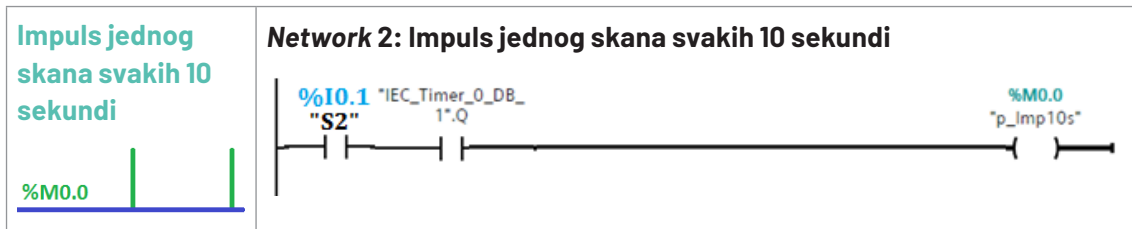
Vrijednost (PT) parametra je 10000.



Slika 4.26. Network1 za Zadatak 3

Po uključenju tajmer svoj izlaz Q postavlja u vrijednost 1 kada odbroji zadano vrijeme. U idućem skanu tajmer ponovo počinje brojanje, a njegov izlaz Q postavlja se u vrijednost 0.

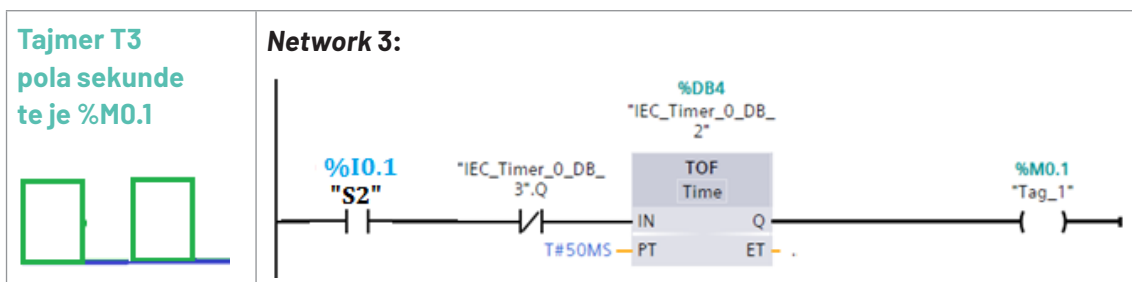
Na isti način, postavlja se i %M0.0.



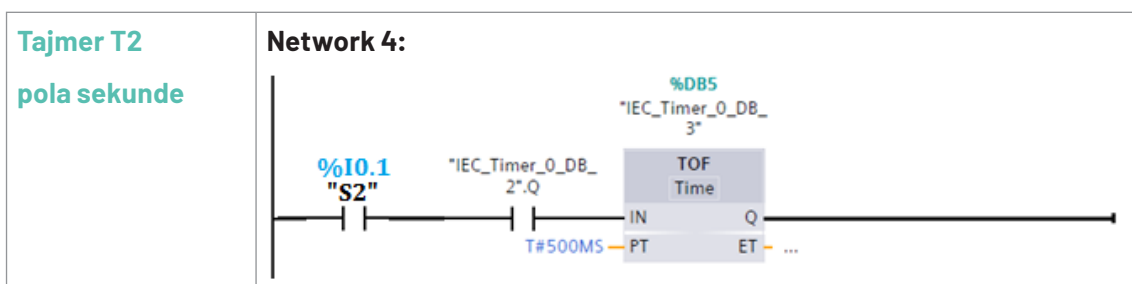
Slika 4.27. Network 2 za Zadatak 3

%M0.0 postavlja se u vrijednost 1 svakih 10 sekundi u trajanju od 1 skana.

Sljedeće dvije slike (slika 4.16 i slika 4.17) prikazuju postavljanje dvaju komplementarnih tajmera u trajanju od pola sekunde svaki. Služe za postavljanje treptajućeg svjetla u ritmu pola sekunde aktivno, pola sekunde neaktivno.



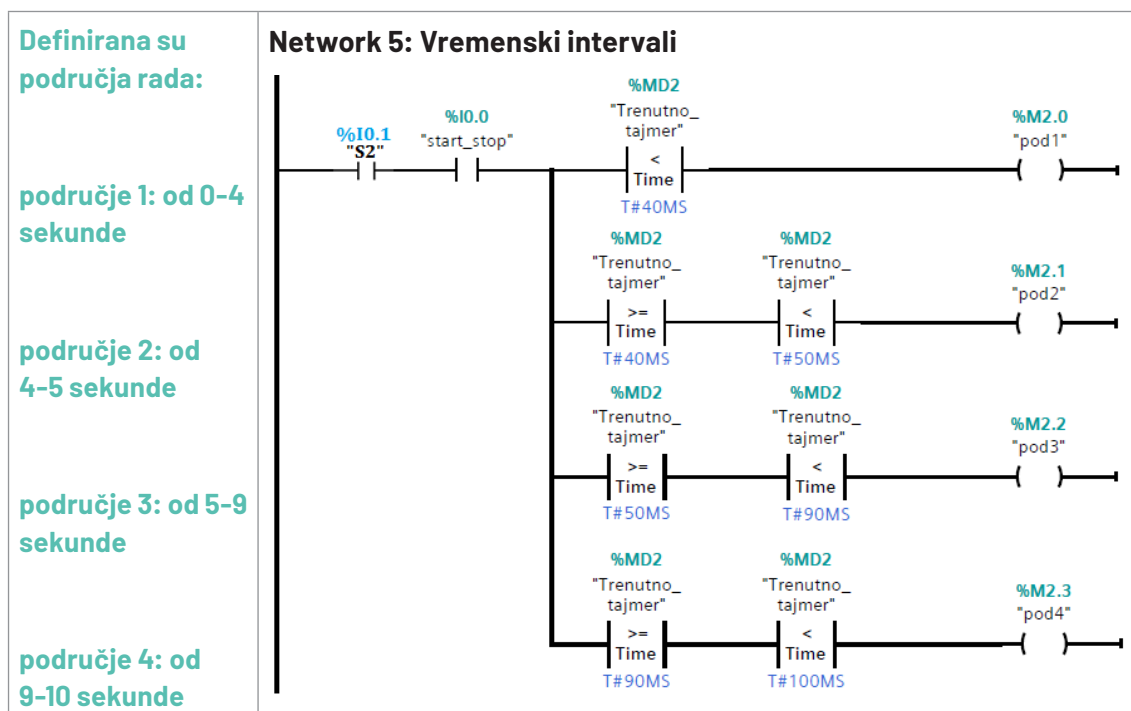
Slika 4.28. Network 3 za Zadatak 3



Slika 4.29. Network 4 za Zadatak 3

Prethodne dvije mreže napisane su za potrebno treptajuće svjetlo. Signal %M0.1 mijenja svoje stanje u skladu s tajmerima. Ritam signal-pauza je 0,5s – 0,5s.

U skladu s dijagramom uključenja-isključenja definirana su područja rada.

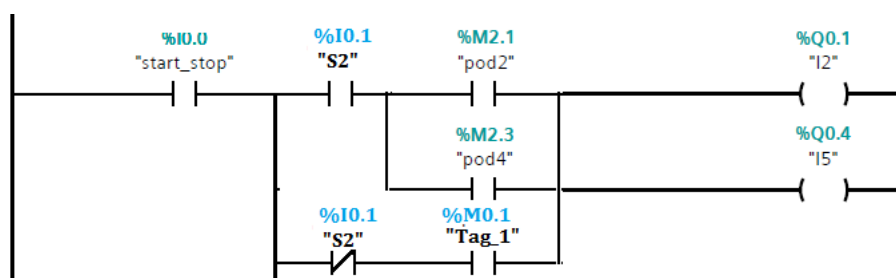


Slika 4.30. Network 5 za Zadatak 3

Područja rada definirana su po vremenskom dijagramu:

- područje 1 od 0 do 4 sec
  - područje 2 od 4 do 5 sec
  - područje 3 od 5 do 9 sec
  - područje 4 od 9 do 10 sec
- postavljenog tajmera.

U skladu s dijagramom uključenja-isključenja postavljaju se izlazi.

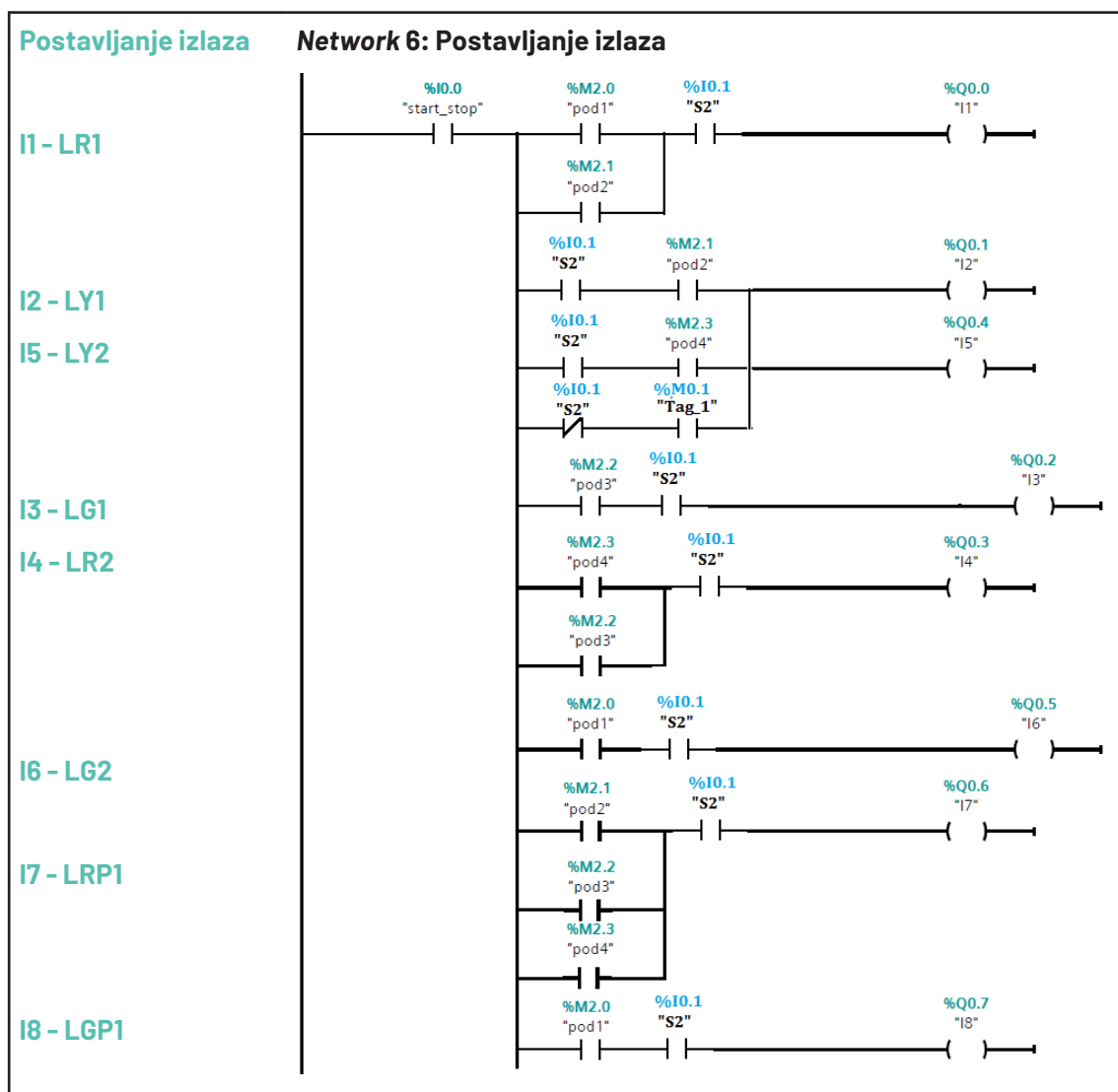


Slika 4.31. Network 6 (detalj) za Zadatak 3

Na prethodnoj slici prikazan je detalj mreže na koju se postavljaju izlazi. Žuta svjetla na semaforima rade na sljedeći način:

- ako je prekidač S2=1, rade u područjima 2 i 4
- ako je prekidač S2=0, trepere u ritmu 0.5s - 0.5s.

Neki izlazi postavljaju se unutar više područja. Npr. u području 1 i 2, odnosno od prve do pete sekunde, pali se crveno svjetlo u smjeru 1.

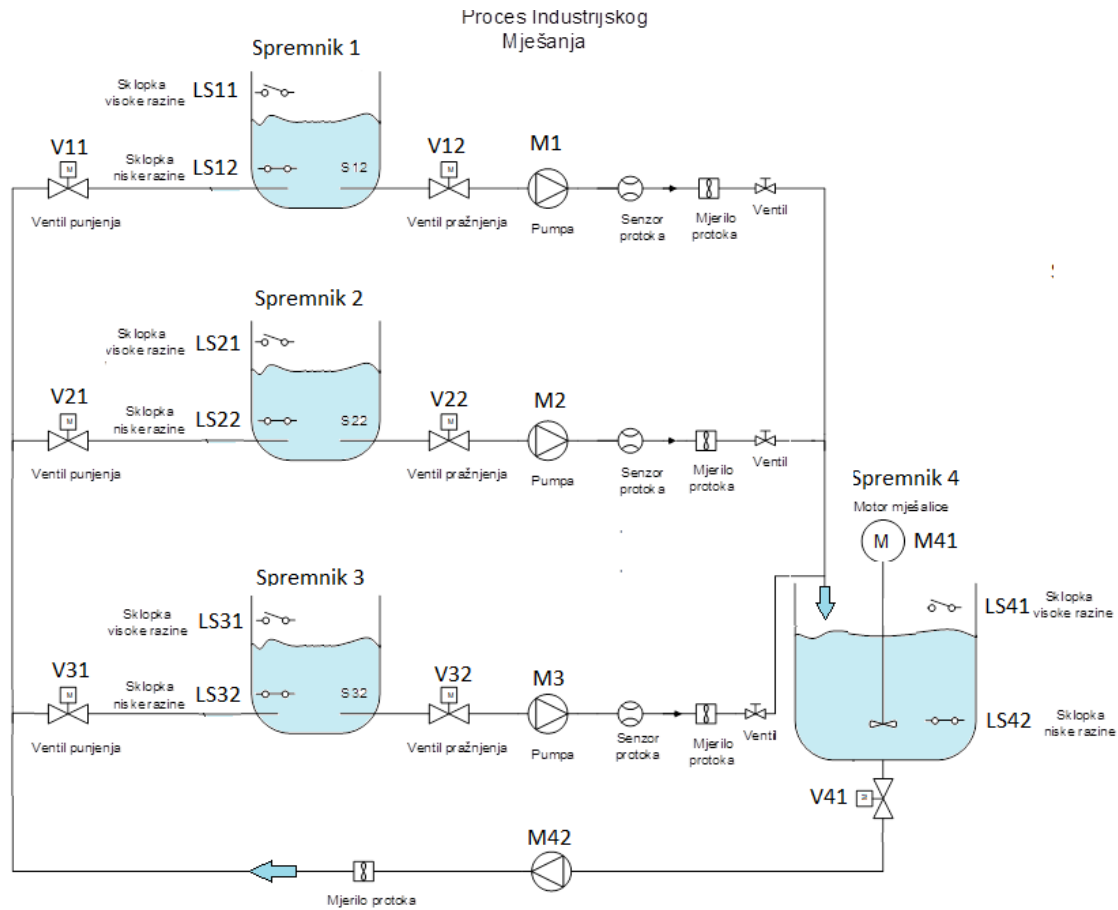


Slika 4.32. Network 6 za Zadatak 3

## 4.4. Zadatak 4: Mjerenje protoka

### 4.4.1. Zadatak

Impulsni mjeri a  protoka teku ina generira niz impulsa  iji je broj u jedinici vremena proporcionalan protoku. Potrebno je izmjeriti broj impulsa.



Slika 4.33. Mjerenje protoka – prikaz za Zadatak 4

- dozvolu pokretanja automatskog ciklusa spojiti na IO.0
- razina vode u spremniku kontrolira se pomo u sklopke visoke (LSx1) i niske razine (LSx2)
- ventil V1 (V11, V21, V31) uklju uje se kada je ispunjen uvjet da je razina vode ispod sklopke LSx2, a isklju uje kada razina vode dosegne sklopku LSx1
- pumpa (M1, M2, M3) se uklju uje kada razina vode aktivira sklopku LSx1., a isklju uje kada razina vode aktivira sklopku LSx2
- valja onemogu iti istovremeno aktiviranje ventila Vx1 i pumpe (M1, M2, M3)
- valja ugraditi vremensku zadršku pri pokretanju ventila Vx2 i pumpe (M1, M2, M3).

## 4.4.2. Rješenje

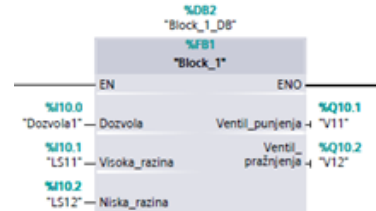
Ovo je samo jedno od mogućih rješenja.

Tri spremnika rade na isti način pa se taj dio programira kao funkcijski blok koji za svoje ulazne parametre ima:

- dozvolu rada
- visoku i nisku razinu spremnika.

Funkcijski blok kao izlazne parametre ima:

- ventil punjenja
- ventil pražnjenja.

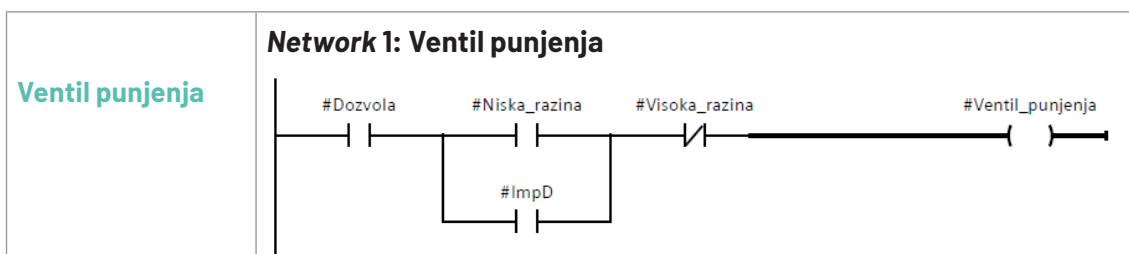


Funkcijski blok koji se poziva FB1 „Block\_1“:

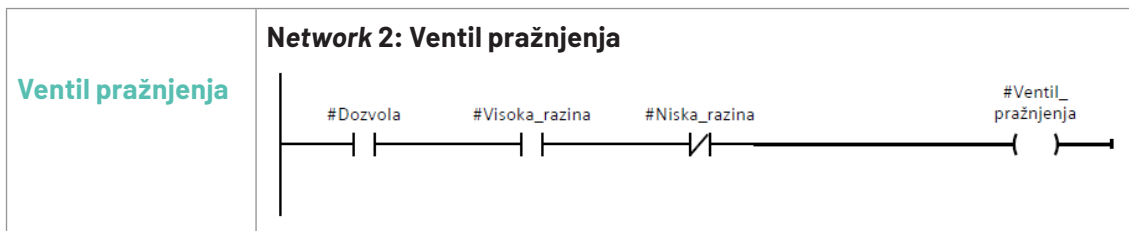
### Block\_1 [FB1]

Name	Data type	Offset	Default value	Retain	Accessi-ble from HMI	Visible in HMI	Comment
▼ Input							
Dozvola	Bool		false	Non-retentive	True	True	
Visoka_razina	Bool		false	Non-retentive	True	True	
Niska_razina	Bool		false	Non-retentive	True	True	
▼ Output							
Ventil_punjenja	Bool		false	Non-retentive	True	True	
Ventil_praznjenja	Bool		false	Non-retentive	True	True	
InOut							
▼ Static							
pomD	Bool		false	Non-retentive	True	True	
ImpD	Bool		false	Non-retentive	True	True	
Temp							

Slika 4.34. Rješenje – FB Zaglavlje bloka za Zadatak 4

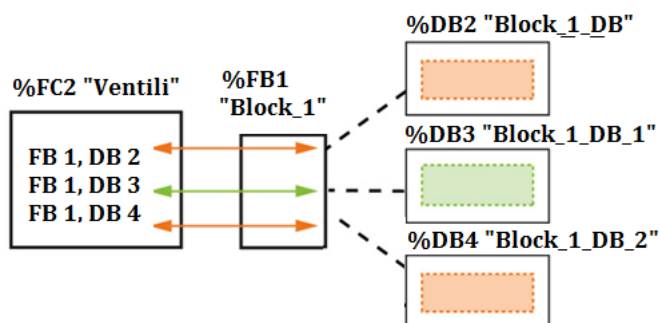


Slika 4.35. Rješenje – FB Network 1 za Zadatak 4



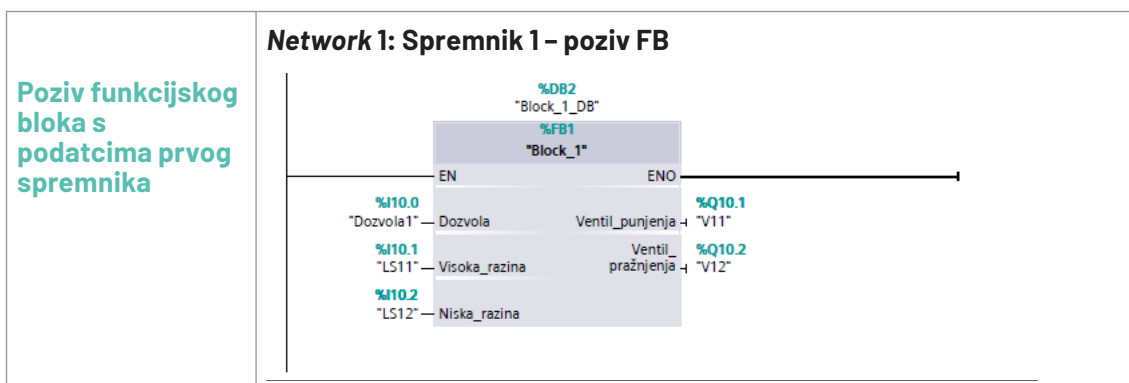
Slika 4.36. Rješenje - FB Network 2 za Zadatak 4

Slijedi funkcija iz koje se poziva funkcijski blok na način prikazan na slici:

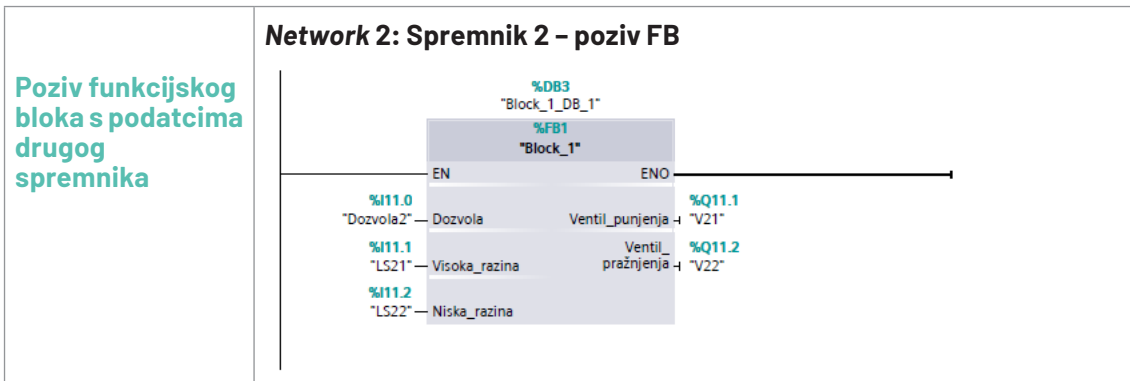


Slika 4.37. Rješenje - FC - poziv FB1 za Zadatak 4

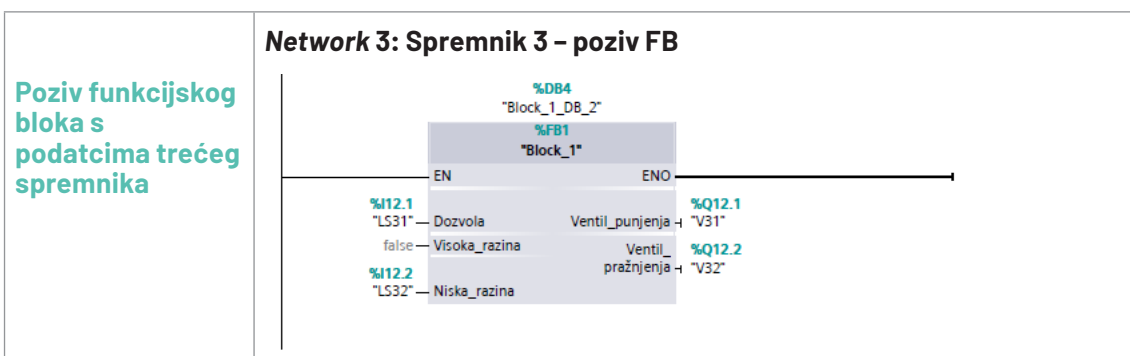
Na ulaz funkcijskog bloka dovode se signali svakog pojedinog spremnika, a na njegovom izlazu, za svaki pojedini spremnik, postavljaju se digitalni izlazi za pripadajući ventil pražnjenja i ventil punjenja.



Slika 4.38. Rješenje - FC Network 1 za Zadatak 4

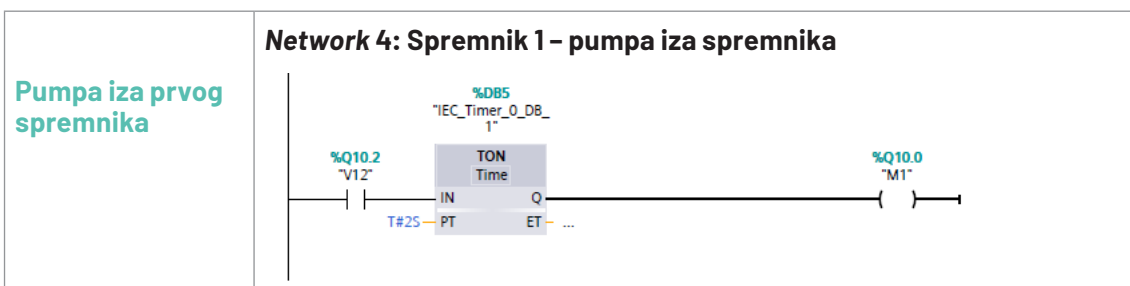


Slika 4.39. Rješenje – FB Network 2 za Zadatak 4



Slika 4.40. Rješenje – FB Network 3 za Zadatak 4

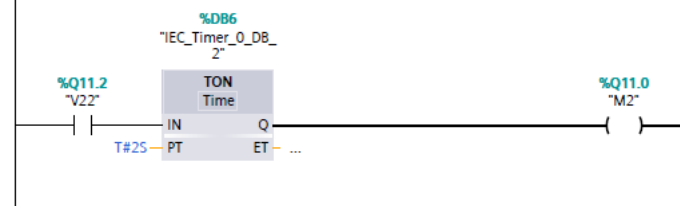
Nakon obrađenih podataka za svaki spremnik, slijede tri mreže u kojima se programira kašnjenje uključanja pumpe (2 sekunde) nakon ventila za pražnjenje svakog pojedinačnog spremnika.



Slika 4.41. Rješenje – FB Network 4 za Zadatak 4

### Pumpa iza drugog spremnika

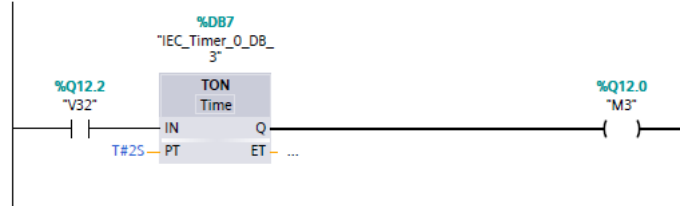
#### Network 5: Spremnik 2 – pumpa iza spremnika



Slika 4.42. Rješenje – FB Network 5 za Zadatak 4

### Pumpa iza trećeg spremnika

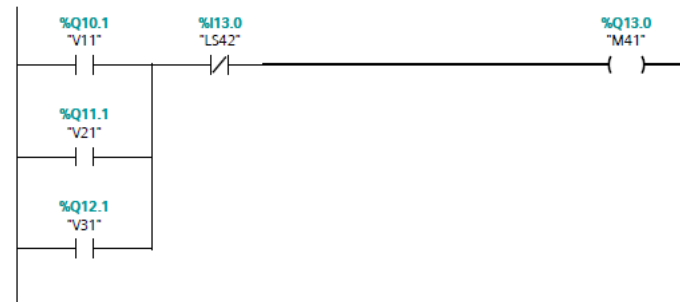
#### Network 6: Spremnik 3 – pumpa iza spremnika



Slika 4.43. Rješenje – FB Network 6 za Zadatak 4

### Glavna pumpa radi ako je uključen bar jedan od ventila punjenja (V11 ili V21 ili V31), a spremnik 4 je iznad niske razine

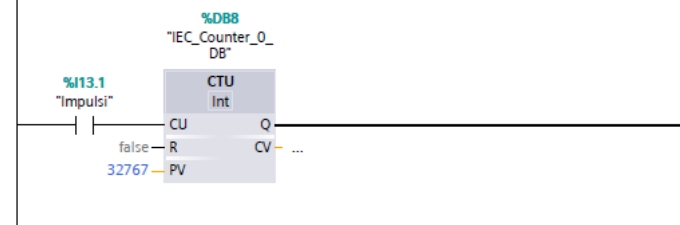
#### Network 7: Glavna pumpa



Slika 4.44. Rješenje – FB Network 7 za Zadatak 4

### Brojenje impulsa brojila protoka vode

#### Network 8: Brojilo protoka vode



Slika 4.45. Rješenje – FB Network 8 za Zadatak 4

## Popis slika

Slika 1.1. Modularni PLC različitih proizvođača.....	11
Slika 1.2. Kompaktni PLC tvrtke SIEMENS.....	12
Slika 1.3. Načelna blok-shema PLC-a .....	12
Slika 1.4. Jedan skan ciklus.....	14
Slika 1.5. Modul digitalnih ulaza.....	15
Slika 1.6. Modul digitalnih izlaza .....	16
Slika 1.7. <i>Sinking</i> i <i>sourcing</i> .....	17
Slika 1.8. PLC kanal kao strujni ponor.....	18
Slika 1.9. PLC kanal kao strujni izvor .....	18
Slika 1.10. PNP senzor.....	19
Slika 1.11. NPN senzor .....	19
Slika 1.12. Modul analognih ulaza 4-20mA .....	20
Slika 1.13. Modul analognih ulaza 0-10V .....	20
Slika 1.14. PLC S-1200 i primjer povezivanja senzora i aktuatora.....	21
Slika 1.15. Komponente SIMATIC S7-1200 sustava.....	22
Slika 1.16. Opcije povezivanja na PLC S7-1200 .....	24
Slika 1.17. Pregled PLC modula .....	25
Slika 1.18. Ljestvičasti dijagram LAD.....	26
Slika 1.19. Dijagram funkcijskih blokova FBD.....	27
Slika 1.20. Strukturirani upravljački jezik SCL.....	27
Slika 1.21. Izvršenje korisničkog programa .....	28
Slika 1.22. Primjer poziva istog bloka s različitim parametrima .....	29
Slika 2.1. Memorijska područja za operande .....	30
Slika 2.2. Clock memorijski bajt.....	33
Slika 2.3. Putanja signala od tipkala do procesne slike ulaza.....	34
Slika 2.4. Putanja signala od procesne slike izlaza do lampice .....	34
Slika 2.5. Raspored bitova, bajtova, riječi i dvostrukih riječi.....	35
Slika 2.6. Struktura BYTE, WORD i DWORD.....	38
Slika 2.7. Struktura za BCD 16-bitni i BCD 32-bitni.....	38
Slika 2.8. Struktura za USINT, UINT i UDINT .....	38
Slika 2.9. Struktura za SINT, INT i DINT .....	39
Slika 2.10. Struktura za REAL i LREAL .....	39
Slika 2.11. Struktura za CHAR .....	40
Slika 2.12. Pulsni tajmer .....	41
Slika 2.13. Tajmer po uključenju .....	41
Slika 2.14. Tajmer po isključenju.....	42
Slika 2.15. Retentivni tajmer.....	42
Slika 2.16. IEC brojač DB.....	43
Slika 2.17. Brojač CTU.....	44
Slika 2.18. Brojač CTD.....	45
Slika 2.19. Brojač CTUD .....	45
Slika 3.1. Ljestvičaste sheme .....	46

Slika 3.2. Struktura programskog bloka .....	47
Slika 3.3. Princip rada NO kontakta .....	49
Slika 3.4. Princip rada NC kontakta .....	50
Slika 3.5. Serijski spoj .....	50
Slika 3.6. Paralelni spoj .....	51
Slika 3.7. Serijsko-paralelni spoj .....	52
Slika 3.8. Kombinirani serijsko-paralelni spoj .....	52
Slika 3.9 T-grananje .....	53
Slika 3.10. Negacija rezultata logičke operacije .....	54
Slika 3.11. Svitci .....	54
Slika 3.12. Standardni i negativni svitak .....	55
Slika 3.13. Set i reset svitak .....	55
Slika 3.14. Pulsni izlaz .....	56
Slika 3.15. IEC tajmer .....	56
Slika 3.16. Primjer aritmetičkih funkcija u programskom jeziku LAD .....	59
Slika 3.17. Funkcija Jump .....	60
Slika 3.18. Funkcijski blok Ventil .....	61
Slika 3.19. Poziv funkcijskog bloka Ventil za Ventil 1 .....	61
Slika 3.20. Poziv funkcijskog bloka Ventil za Ventil 2 .....	62
Slika 3.21. Poziv funkcijskog bloka .....	62
Slika 3.22. Poziv funkcije uvjetno .....	62
Slika 3.23. Poziv funkcije bezuvjetno .....	63
Slika 4.1. Zadatak 1j .....	65
Slika 4.2 Rješenje 1a .....	65
Slika 4.3. Rješenje 1b – funkcija pamćenja .....	66
Slika 4.4. Rješenje 1b – bistabil .....	66
Slika 4.5. Rješenje 1c .....	66
Slika 4.6. Rješenje 1d .....	67
Slika 4.7. Rješenje 1e .....	68
Slika 4.8. Rješenje 1f .....	68
Slika 4.9. Rješenje 1g .....	69
Slika 4.10. Rješenje 1h .....	69
Slika 4.11. Rješenje 1i .....	70
Slika 4.12. Rješenje 1j .....	70
Slika 4.13. Vremenski dijagram uključenja izlaza za Zadatak 2 .....	72
Slika 4.14. Rješenje A – Network 1 za Zadatak 2 .....	73
Slika 4.15. Rješenje A – Network 2 za Zadatak 2 .....	73
Slika 4.16. Rješenje B – Network 1 za Zadatak 2 .....	74
Slika 4.17. Rješenje B – Network 2 za Zadatak 2 .....	74
Slika 4.18. Rješenje C – Network 1 za Zadatak 2 .....	75
Slika 4.19. Rješenje C – Network 2 za Zadatak 2 .....	75
Slika 4.20. Rješenje A,B,C – aktiviranje brojača za Zadatak 2 .....	76
Slika 4.21. Rješenje A,B,C – postavljanje izlaza za Zadatak 2 .....	77
Slika 4.22. Rješenje S1=0 za Zadatak 3 .....	79
Slika 4.23. Rješenje S1=1 i S2=1 za Zadatak 3 .....	79

Slika 4.24. Vremenski dijagram uključenja i isključenja svjetala – grafički za Zadatak 3.....	81
Slika 4.25. Rješenje S1=1 i S2=0 za Zadatak 3 .....	82
Slika 4.26. Network1 za Zadatak 3.....	82
Slika 4.27. Network 2 za Zadatak 3.....	83
Slika 4.28. Network 3 za Zadatak 3 .....	83
Slika 4.29. Network 4 za Zadatak 3 .....	83
Slika 4.30. Network 5 za Zadatak 3 .....	84
Slika 4.31. Network 6 (detalj) za Zadatak 3.....	84
Slika 4.32. Network 6 za Zadatak 3 .....	85
Slika 4.33. Mjerenje protoka – prikaz za Zadatak 4 .....	86
Slika 4.34. Rješenje – FB Zaglavlje bloka za Zadatak 4 .....	87
Slika 4.35. Rješenje – FB Network 1 za Zadatak 4 .....	87
Slika 4.36. Rješenje – FB Network 2 za Zadatak 4 .....	88
Slika 4.37. Rješenje – FC – poziv FB1 za Zadatak 4 .....	88
Slika 4.38. Rješenje – FC Network 1 za Zadatak 4 .....	88
Slika 4.39. Rješenje – FB Network 2 za Zadatak 4 .....	89
Slika 4.40. Rješenje – FB Network 3 za Zadatak 4 .....	89
Slika 4.41. Rješenje – FB Network 4 za Zadatak 4 .....	89
Slika 4.42. Rješenje – FB Network 5 za Zadatak 4 .....	90
Slika 4.43. Rješenje – FB Network 6 za Zadatak 4 .....	90
Slika 4.44. Rješenje – FB Network 7 za Zadatak 4 .....	90
Slika 4.45. Rješenje – FB Network 8 za Zadatak 4 .....	90

## Popis tablica

Tablica 2.1. ID operanda i apsolutno adresiranje .....	35
Tablica 2.2. Raspon dopuštenih vrijednosti konstanti .....	36
Tablica 2.3. IEC tajmeri.....	41
Tablica 2.1. ID operanda i apsolutno adresiranje .....	43
Tablica 3.1. Tipovi programskih naredbi LAD.....	48
Tablica 3.2. Relejni kontakti .....	48
Tablica 3.3. Funkcija - usporedba .....	57
Tablica 3.4. Funkcija - premještanje.....	57
Tablica 3.5. Funkcija - konverzija.....	58
Tablica 3.6. Aritmetičke funkcije u programskom jeziku LAD .....	58
Tablica 4.1. Tablica ulaza i izlaza za Zadatak 1.....	64
Tablica 4.2. Tablica ulaza i izlaza za Zadatak 2 .....	71
Tablica 4.3. Tablica ulaza i izlaza za Zadatak 3 .....	78
Tablica 4.4. Vremenski dijagram uključenja i isključenja svjetala – tablično za Zadatak 3.....	80

## Popis korištenih kratica

**ASCII** = 7-bitni kôd za prikaz alfanumeričkih znakova, znakova interpunkcije i kontrolnih znakova

(engl. **American Standard Code for Information Interchange**)

**ASI** = ASI interface (engl. **Actuator Sensor Interface**)

**BCD** = binarni brojevni sistem u kojem je svaka decimalna znamenka od 0 do 9 prikazana s 4 bita (engl. **Binary Coded Decimal**)

**CB** = kartica za komunikaciju (engl. **Communication Boards**)

**CM** = komunikacijski modul (engl. **Communication Modules**)

**CP** = komunikacijski procesor (engl. **Communication Processor**)

**CPU** = centralna procesorska jedinica (engl. **Central Processing Unit**)

**CTD** = brojač naniže (engl. **Counter Down**)

**CTU** = brojač naviše (engl. **Counter Up**)

**CTUD** = brojač naviše/naniže (engl. **Counter Up and Down**)

**CV** = trenutna vrijednost brojača (engl. **Current count Value**)

**DB** = podatkovni blok (engl. **Data Block**)

**ET** = trenutna vrijednost tajmera (engl. **Elapsed Time**)

**FB** = funkcijski blok (engl. **Function Block**)

**FBD** = dijagram funkcijskih blokova (engl. **Function Block Diagram**)

**FC** = funkcija (engl. **Function**)

**HMI** = operacijski panel (engl. **Human Machine Interface**)

**HSC** = brzi brojač (engl. **High Speed Counter**)

**I** = ulazni signal (engl. **Input**)

**IEC** = IEC standard (engl. **International Electrotechnical Commission**)

**IN** = signal na ulazu (engl. **Input**)

**LAD** = ljestvičasti dijagram (engl. **Ladder Logic**)

**MMC** = mikromemorijska kartica (engl. **Micro Memory Card**)

**NO** = normalno otvoren (engl. **normally open**)

**NC** = normalno zatvoren (engl. **normally closed**)

**OB** = organizacijski blok (engl. **Organization Block**)

**PLC** = programabilni logički kontroler (engl. **Programmable Logic Controller**)

**PT** = trajanje tajmera (engl. **Preset Timer**)

**PV** = zadana vrijednost brojača (engl. **Preset Value**)

**Q** = izlazni signal (engl. *output*)

**R** = reset (engl. **Reset**)

**SCL** = strukturirani upravljački jezik (engl. **Structured Control Language**)

**SB** = signalna kartica (engl. **Signal Board**)

**SM** = signalni modul (engl. **Signal Modules**)

**TOF** = tajmer po isključenju (engl. **Timer OFF delay**)

**TON** = tajmer po uključenju (engl. **Timer ON delay**)

**TONR** = akumulirajući tajmer (engl. **Timer ON delay retentive**)

**TP** = pulsni tajmer (engl. **Timer Pulse**)

## Rječnik pojmova

A

### **Adresa.**

Lokacija u memoriji računala na kojoj su smješteni podatci

### **Adresno polje.**

Niz od 8, 16, 32 ili 64 bita kojom je određena pozicija u memoriji.

### **Algoritam.**

Set procedura koje se upotrebljavaju za rješavanje određenog zadatka.

### **Alfanumerički kôd.**

Niz znakova koji se sastoji od slova, brojeva i/ili specijalnih znakova.

### **Analogni uređaj.**

Uređaj koji je namijenjen za obradu analognog signala.

### **Analogni signal.**

Signal koji ima kontinuiranu, neprekinutu i promjenjivu vrijednost.

### **Analogno-digitalni pretvarač**

Sklop dizajniran za pretvaranje analognog električnog signala u višebitnu binarnu riječ.

### **AND**

Logička operacija koja zahtijeva da svi ulazni parametri budu u logičkoj jedinici kako bi njen izlazni parametar postao logička jedinica.

### **Apsolutno adresiranje.**

Pri adresiranju koristi se operand koji se sastoji od ID operanda i memorijske adrese.

B

### **Bajt**

Tip podatka veličine 8 bita.

### **BCD kodirani brojevi**

Binarni kod kojim se kodiraju samo numeričke znamenke 0 – 9.

### **Bit**

Najmanja jedinica binarne informacije. Može imati vrijednost 0 ili 1.

### **BOOL**

Tip podatka koji predstavlja vrijednost jednog bita.

C

### **Centralna procesorska jedinica.**

Dio PLC-a odgovoran za čitanje ulaza, izvršavanje programa i postavljanje izlaza.

### **CHAR.**

Tip podataka CHAR predstavlja jedan znak koji se sprema u ASCII formatu.

**Ciklički prekid.**

Prekidom upravljano izvođenje programa, događaj koji se odvija u definiranim intervalima.

**Clock memorijski bitovi.**

Nalaze se na programiranom bajtu i daju periodički signal točno određene frekvencije.

D

**Dekadski brojevni sustav.**

Brojevni sistem s bazom 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 i 0; koriste se za prikaz svih vrijednosti. Svaka znamenka pozicionirana je na mjestu određene težinske vrijednosti od 1, 10, 100, 1000, ...

**Dijagram toka.**

Grafički prikaz definicije zadatka ili rješenja problema.

**Diskretna podatkovna točka.**

Podatkovna točka sa samo dva stanja: uključeno i isključeno.

**DWORD**

Tip podatka veličine 32 bita.

F

**Firmver** (engl. *firmware*)

Softver koji omogućava rad hardvera. Koristi se za pokretanje korisničkih programa.

**Floating point.**

Format podatka.

**Funkcijski blok.**

Blok koji se koristi za izradu programa, ima memoriju u koju sprema lokalne podatke.

H

**Hardverski prekid.**

Prekidom upravljano izvođenje programa koje postavlja zahtjev za brzim odgovorom.

I

**ID operanda.**

Definira tip podatka koji je dodijeljen operandu.

**IEC61131.**

Standard koji definira glavne karakteristike PLC uređaja te povezane periferne opreme, način komunikacije između PLC-ova i druge opreme te pravila za korištene programske jezike. Dio III ovog standarda specificira sintaksu i semantiku unificiranog skupa programskih jezika za PLC-ove.

**Indirektno adresiranje.**

Indirektno ili neizravno adresiranje koristi se kada se isti programski dijelovi koriste višestruko, ali s različitim operandima.

**Instanc blok.**

Podatkovni blok pridružen funkcijskom bloku.

K

**Komunikacijski modul.**

Modul preko kojeg PLC, putem različitih vrsta sabirnica, komunicira s različitim „inteligentnim“ uređajima.

L

**LREAL.**

64-bitni broj, sastoji se od tri komponente: predznaka, 11-bitnog eksponenta i 52-bitne mantise.

LJ

**Ljestvičasti dijagram.**

Grafički programski jezik temeljen na relejnoj logici.

M

**Mrežni protokol.**

Mrežni komunikacijski protokoli kojima PLC-ovi komuniciraju.

**Memorija za učitavanje.**

Nalazi se na memorijskoj kartici i koristi za smještaj upravljačkog programa.

N

**NPN senzor.**

Senzor koji je tipa strujni ponor.

O

**Organizacijski blok.**

Obvezan blok koji predstavlja vezu između operativnog sustava i korisničkog programa.

P

**Periferni ulazi.**

Memorijsko područje na ulaznim modulima. Izravno sučelje s procesom ili strojem, očitavanje stanja sa senzora, preklopki, tastera.

**Periferni izlazi.**

Memorijsko područje na izlaznim modulima. Izravno sučelje s procesom ili strojem, postavlja nje komandi prema signalizaciji, motorima, ventilima.

**Procesna slika ulaza.**

Memorijsko područje u koje se upisuje status svih ulaznih modula.

**Procesna slika izlaza.**

Memorijsko područje iz kojeg se upisuju statusi svih izlaznih modula.

**PNP senzor.**

Senzor koji je tipa strujni izvor.

**Podatkovni blok.**

Dio memorije u koji se pohranjuju varijable potrebne za izvršenje programa.

R

**Radna memorija.**

Memorija integrirana je u CPU koja se koristi za smještaj podataka potrebnih za izvršenje programa.

**REAL.**

32-bitni broj, sastoji se od tri komponente: predznaka, 8-bitnog eksponenta i 23-bitne mantise.

**Retentivna memorija.**

integrirana je u CPU i koristi se za čuvanje određenih podataka pri nestanku napajanja i ponovno pokretanje CPU-a.

**Retentivni memorijski bitovi.**

Memorijski bitovi koji zadržavaju svoje stanje čak i nakon prestanka napajanja. Područje unutar kojeg se nalaze definira korisnik unutar tablice tagova.

S

**Simboličko adresiranje.**

Operandu se dodjeljuje unikatno ime.

**Sistemska memorija.**

Memorija integrirana u CPU i koristi se za sliku ulaza, sliku izlaza, interne rezultate, za ulaze i izlaze, međurezultate, za čuvanje privremenih podataka te za tajmere i brojače.

**Sistemske memorijske bitove.**

Nalaze se na programiranom bajtu. Ovaj memorijski bajt kontrolira operativni sustav CPU-a.

**Sistemske tipove podataka.**

Sistemske tipove podataka unaprijed su definirani tipove podataka koji se sastoje od fiksnog broja komponenti od kojih svaka ima različite osnovne tipove podataka. Sistemske tipove podataka isporučuju se zajedno s alatom za programiranje i ne mogu se mijenjati.

**Skan.**

Jedan ciklus obrade programa.

**Strujni izvor.**

Uređaj koji šalje struju iz svog upravljačkog priključka na neki drugi uređaj.

**Strujni ponor.**

Uređaj koji prihvaća struju u svoj upravljački priključak.

W

**WORD**

Tip podatka veličine 16 bitova.

## Popis literature

1. INTERNATIONAL STANDARD IEC 61131-3 Second edition 2003-01, IEC
2. Learn-/Training Textbook, Siemens Automation Cooperates with Education (SCE), TIA Portal Modules for Automation System SIMATIC S7-1200 from Version V14 SP1
3. Learn-/Training Document, Siemens Automation Cooperates with Education, (SCE), From Version V14 SP1 TIA Portal Module 031-100, Basics of FC Programming with SIMATIC S7-1200
4. Learn-/Training Document, Siemens Automation Cooperates with Education (SCE) | From Version V14 SP1, TIA Portal Module 031-410, Basics of Diagnostics with SIMATIC S7-1200
5. SIMATIC S7-1200, Getting started with S7-1200



**EduSplit** Obrtna tehnička škola  
Regionalni centar kompetentnosti Split



Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Za više informacija o EU fondovima molimo pogledajte web-stranicu  
Ministarstva regionalnoga razvoja i fondova Europske unije.  
[www.strukturfondovi.hr](http://www.strukturfondovi.hr)