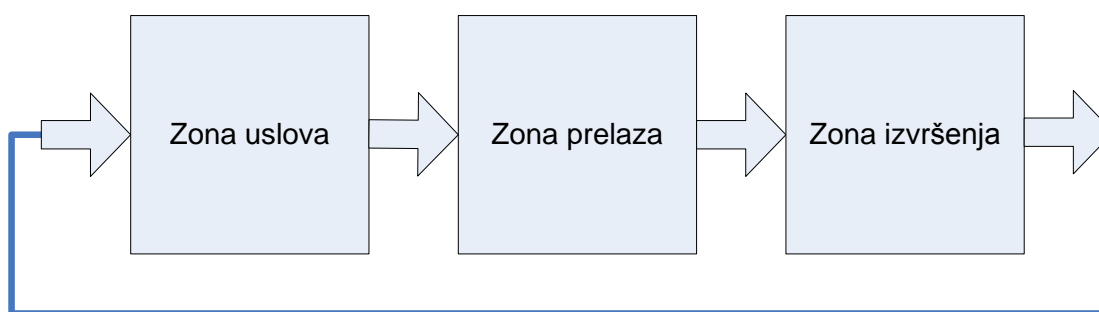


5 Programiranje PLC-ova preko algoritma "State machine"

U praksi se za pisanje programa za PLC-ove najpraktičniji pokazalo primena "state machine" tj. mašine stanja tako da se PLC ponaša kao konačni automat. Osnovna prednost ovog stila programiranja je preglednost i čitljivost koda, kao i mogućnost brze promene funkcionisanja koda.

Struktura koda u obliku dijagrama stanja se može podeliti u tri osnovna dela, kao što se može videti na slici 5.1. Kod se izvršava sekvencijalno, kao što je i uobičajeno kod PLC-ova.



Slika 5.1

Zona uslova predstavlja deo koda kojim se određuje uslovi za prelaz iz trenutnog u buduće stanje. U zoni uslova određuje se funkcija prelaza, buduće stanje zavisi od trenutnog stanja i odgovarajućih ulaza (pri čemu se ovde ne misli samo na fizičke ulaze u PLC, nego i promene tajmera, brojača i ostalih internih elemenata PLC-a mogu biti ulazi koji određuju uslove za prelaz iz jednog u drugo stanje).

Zona prelaza predstavlja deo koda koji pamti i menja trenutno stanje. Ovaj deo koda obavlja funkciju registra stanja konačnog automata. Ovaj deo koda je univerzalan, i minimalno se menja, bez obzira na tip zadatka koji PLC treba da izvrši.

Zona izvršenja podešava stanja izlaza PLC-a. Svi izlazi iz PLC-a su grupisani u ovom delu koda, što olakšava izmene u radu sistema.

5.1 Zona prelaza

Primer koda za zonu prelaza prikazan je na slici 5.2 i slici 5.3. Ovaj deo koda se obično piše kao funkcijski blok, radi lakše implementacije za različite zadatke. Na slici 5.3 je predstavljen kod za automat sa tri stanja, dodavanje novih stanja svodi se na kopiranje prikazanog koda i promenom odgovarajućih ulaza i stanja.

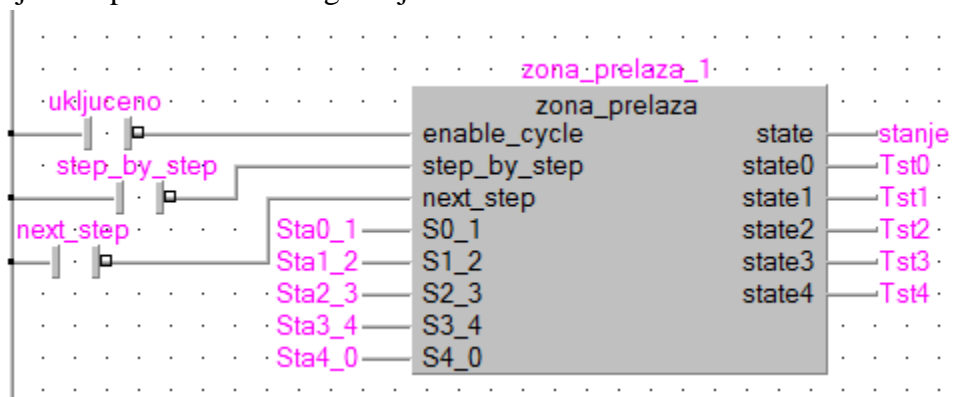
Ovaj funkcijski blok sadrži sledeće promenjive:

Ulazi:

- **Enable_cycle** određuje da li se automat može prebaciti iz trenutnog stanja u buduće, može se posmatrati kao promenjiva koja određuje da li automat radi ili je pauziran/isključen.
- **Step_by_step** - ako je ova promenjiva aktivna, a enable_cycle neaktivan, automat ne može d predje u sledeće stanje sve dok se ne aktivira ulaz next_step. Ovaj režim rada omogućava da se manuelno prelazi iz trenutnog stanja u novo, što je korisno pri testiranju sistema
- **next_step** je ulaz koji omoćava manuelni prelaz u novo stanje, kad se aktivira
- **S0_1, S1_2, S2_3 Sn_0** su ulazi koji, kada se aktiviraju, javljaju automatu da se prebaci iz trenutnog stanja u buduće stanje

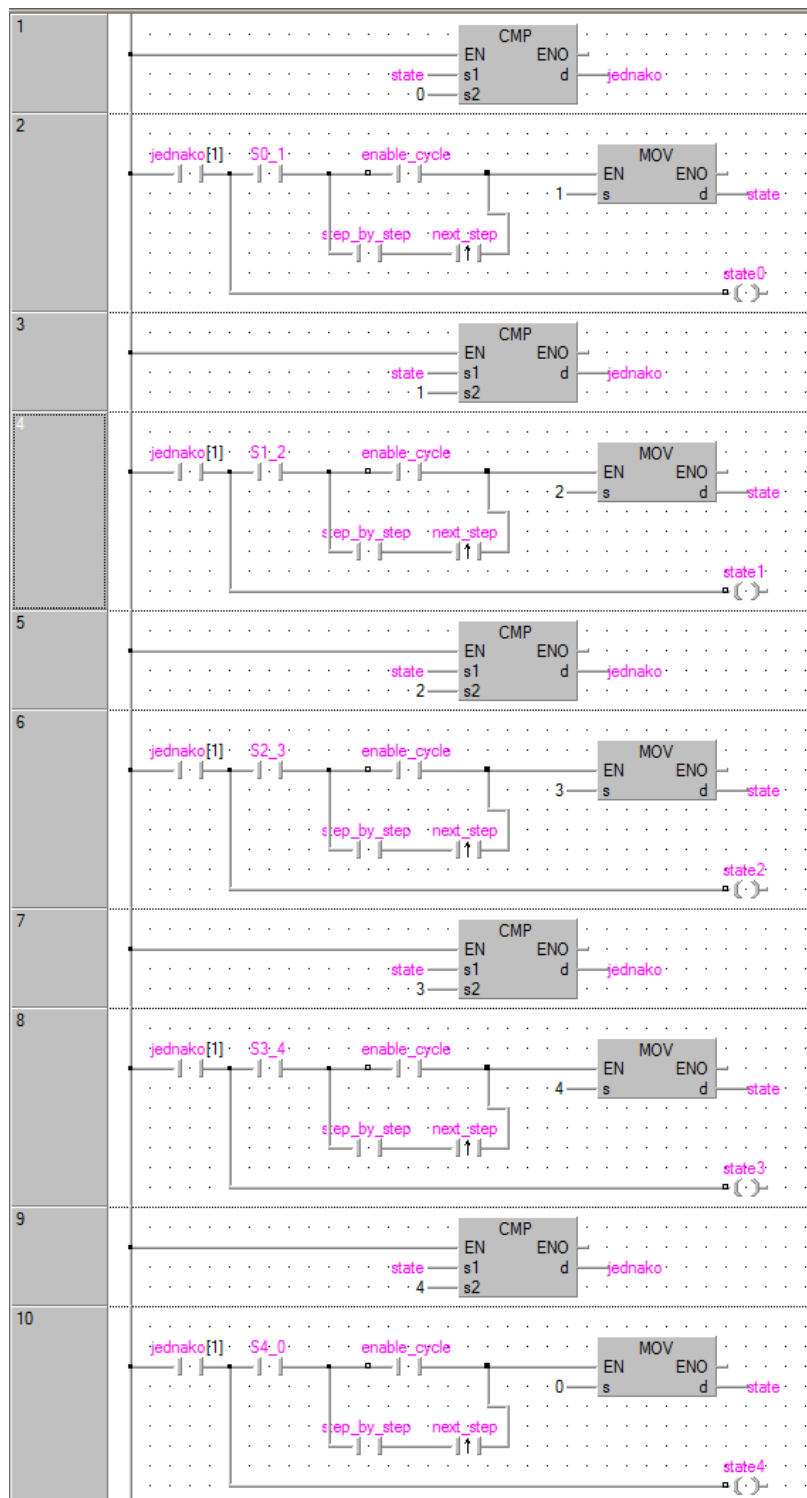
Izlazi:

- **STATE** -int promenjiva u kojoj se čuva trenutno stanje, ova promenjiva može biti i interna, ako se u ostatku automata koriste bool promenjive za proveru trenutnog stanja
- **State0, State1...StateN** bool izlazi koji se koriste u ostatku automata kao bool promenjive za proveru trenutnog stanja



Slika 5.2 Zona prelaza kao instanca funkcijskog bloka

Algoritam rada u zoni prelaza je : preko CMP instrukcije se proverava trenutno stanje(npr. u lestvici 1 proverava se da li je vrednost *state* promenjive jednako 0, tj da li je trenutno stanje jednako nultom stanju). Ako jeste, onda je izlaz CMP instrukcije *jednako[1]* ON i uključuje i odgovarajuć izlaz (npr. *state0* u lestvici 2) . Kada se aktivira neki od ulaza **S0_1, S1_2, S2_3 Sn_0** menja se stanje automata preko instrukcije MOV, npr. promenjiva *state* dobija vrednost 1 u lestvici 2. Logika rada u lestvici 1 i 2 se ponavlja do kraja koda zone prelaza u zavisnost od potrebnog broja stanja.



Slika 5.3 Zona prelaza

5.2 Zona ulaza

U zoni uslova se rešava algoritam rada postavljenog zadatka, proveravaju se odgovarajući ulazi (najčešće spoljašni signali ili stanja tajmera i brojača) u automat i na osnovu toga odlučuje se da

li su ispunjeni uslovi da se pređe u sledeće stanje. predstavlja Kao primer koda sa zonom ulaza rešen je sledeći zadatak:

Potrebno je napisati program za sistem za plastificiranje koja ce obavljati sledeci zadatak:

*1.Kada se pritisne **Start** taster,pokreće se sistem. Kada se pritisne taster **Pokreni_proces** ukreće zagrevanje prostora za plastificiranje*

2.Kada temperatura u prostoru za plastificiranje poraste iznad odredjene granice, eksterni termostat menja stanje na jednom od ulaza PLC-a

3.Nastaviti sa grejanjem sledećih 7 sekundi

4.Ugasiti grejanje i drzati uključen ventilator u sledećih 5 sekundi

5.Ugasiti ventilator i drzati uključeno signalno svetlo u sledećih 10 sekundi

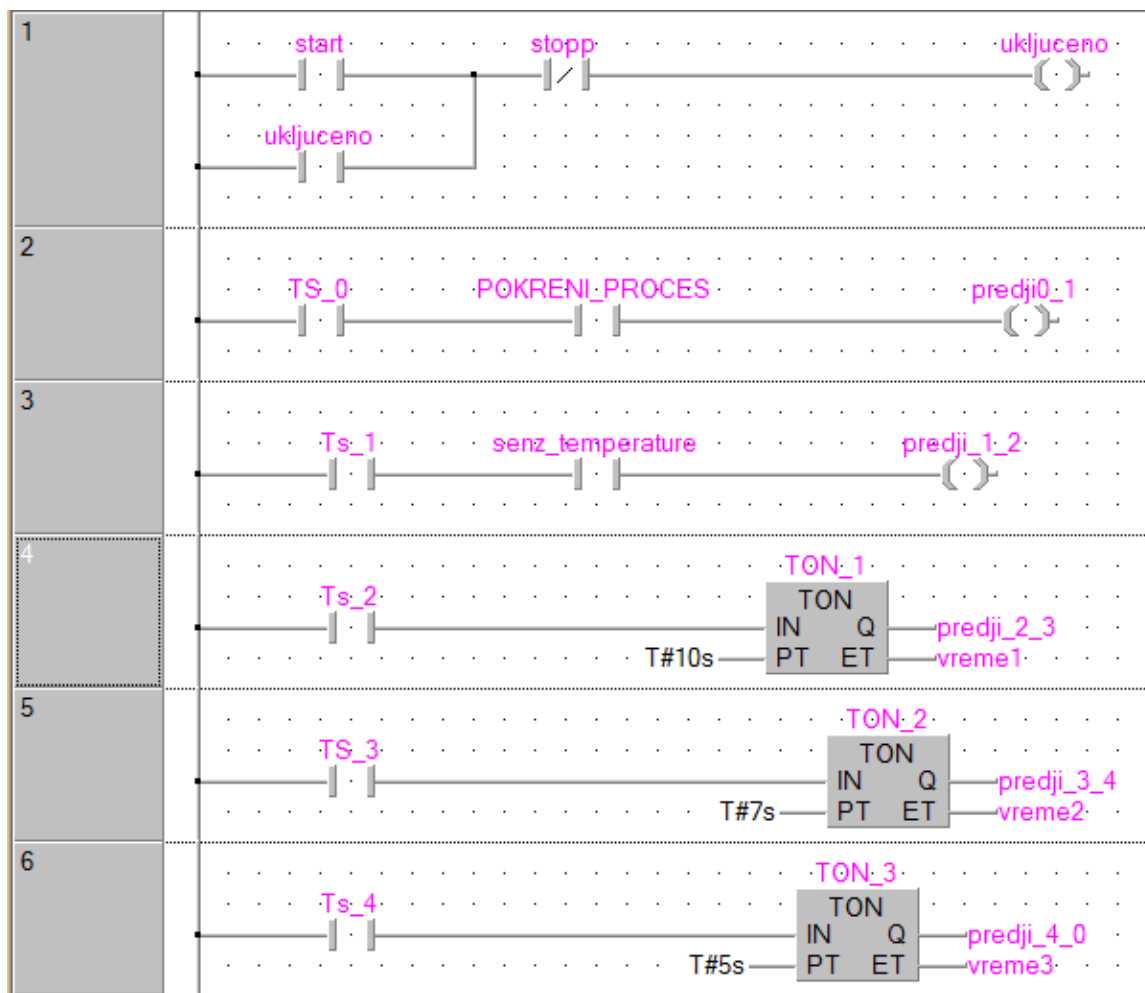
*6.Ugasiti i svetlo, i čekati na **Pokreni_proces** taster da proces krene ispočetka*

Zona ulaza za program koji rešava ovaj zadatak prikazan je na slici 5.4. Na osnovu ulaza u PLC i stanja izlaza internih tajmera PLC-a, menjaju se stanja automata.

Ulazi sistema su:

- Start taster
- Stop taster
- Pokreni_proces
- Senzor_temp

Kada se pokrene sistem pritiskom na Start dugme, automat je u stanju 0 i aktivan je $Ts0$. Sistem čeka na pritisak tastera **Pokreni_proces**, posle čega se stanje preko promenjive $Sta0_1$ menja iz stanja OFF u stanje ON. Kada se uključi ulaz **Senz_temp** automat prelazi u stanje 2. Kada prodje vreme podešeno preko tajmera TON_1 automat prelazi u stanje 3. Kada prodje vreme podešeno preko tajmera TON_2 automat prelazi u stanje 4. Kada prodje vreme podešeno preko tajmera TON_3 automat prelazi u stanje 0, tj vrati se na početak procesa i čeka na pritisak tastera **Pokreni_proces**.



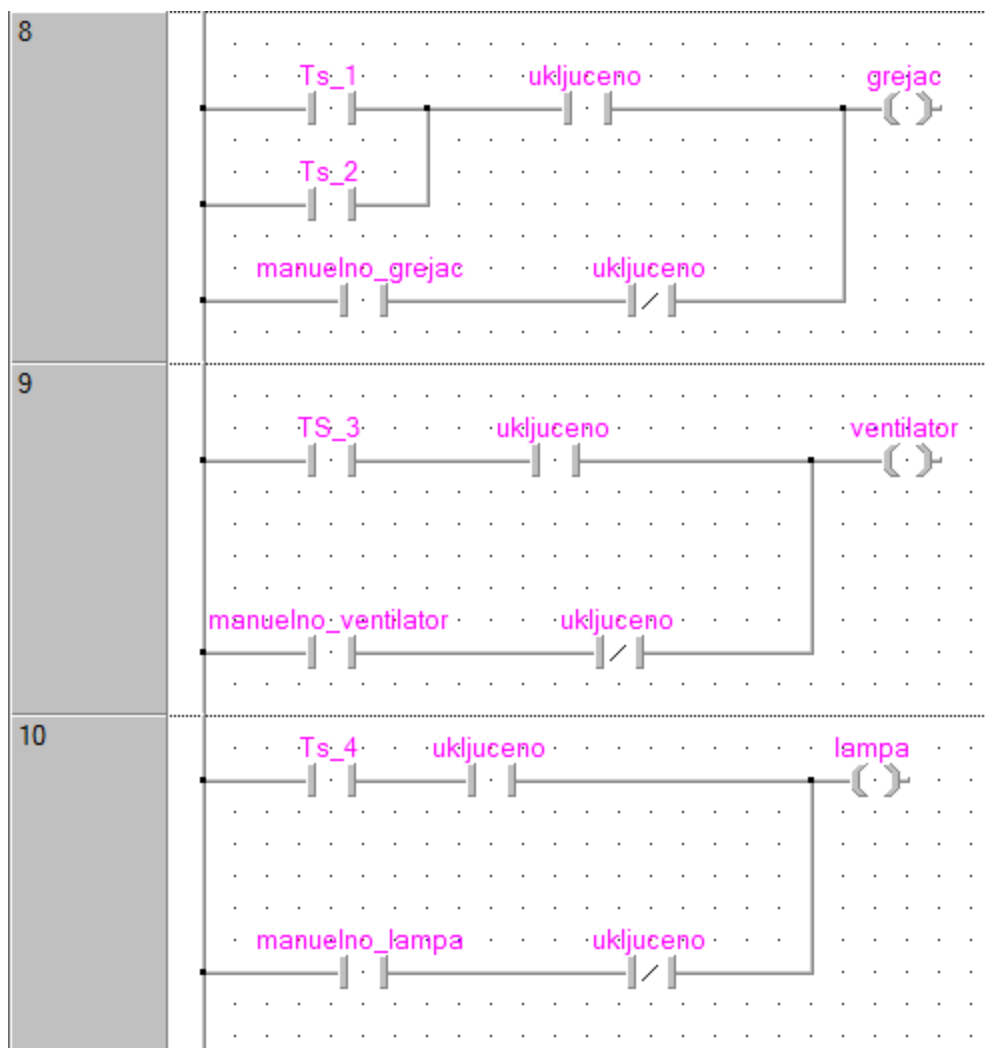
Slika 5.4 Ladder dijagram za zonu uslova

5.3 Zona izvršenja

Kao primer koda sa zonom izvršenja rešeno je upravljanje izlaza prethodnog zadatka. Zona izvršenja je najčešće veoma jednostavna, prema uslovima zadatka proceni se u kom stanju određeni izlaz treba da bude aktivan, i napiše se ladder dijagram koji to izvršava, kao na slici 5.5.

Izlazi sistema

- Grejač
- Ventilator
- Svetlo



Slika 5.5 Ladder dijagram za zonu izvršenja

Iz uslova zadatka i postavke zone uslova jasno je da grejač treba da bude uključen samo u stanjima 1 i 2, ventilator u stanju 3 i svetlo u stanju 4.

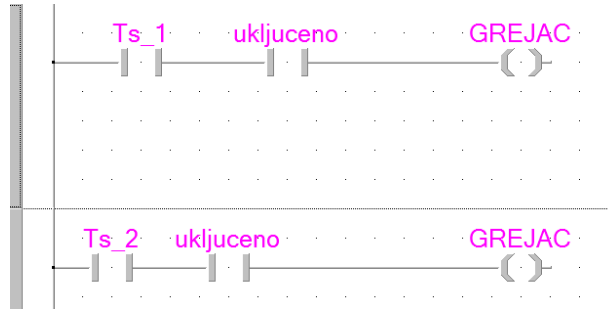
Pomoću promenjive *uključeno* kontrolišemo da izlazi mogu budu aktivni kada je automat aktivan. Pored toga, u praksi se pojavljuje potreba da se prilikom testiranja rada sistema ručno provere svi izlazi u sistemu, tj. da se mogu uključiti i isključiti preko tastera nezavisno od algoritma rada. U tu svrhu se koriste ulazi označeni sa *manuelno* (npr. *manuelno_grejac*), koji utiču na stanje odgovarajućeg izlaza kada je automat neaktivan.

Napomene:

- Nema potrebe da se piše kod za isključivanje određenog izlaza pošto se on automatski ugasi kada prođe stanje u kom je trebao biti aktivan. Npr. grejač je trebao biti ON samo tokom

stanja 1 i 2, što je i obezbeđeno kodom u lestvici 8 na slici 5.5, i nema potrebe da se isključuje u nekoj drugoj lestvici.

- Česta greška je da se isti izlaz pojavljuje u više lestvica koda, npr. da umesto lestvice 8 stavimo dve lestvice, kao na slici 5.6 . U tom slučaju gornja lestvica nema nikakav uticaj na stanje izlaza *grejac*, tj. kao da je ni nema, jer je donja lestvica ponovo podesila stanje izlaza *grejac*, i samim tim je dominantna.



Slika 5.6