



Univerzitet u Kragujevcu

Mašinski fakultet

Odsek: Informatika u inženjerstvu

Predmet: Računarski podržano merenje i upravljanje

Osnove rada sa PLC-om i Touch Panel-om

Profesor:

Dr. Milan Matijević

studenti:

Djurdjević Marko 46/99

Gavrilović Miljan 45/2002

Veljović Ivan 42/2000

Uvod

Ovaj seminarski rad je nastao kao rezultat zajedničkog rada studenata Mašinskog Fakulteta Ivana Veljovića, Miljana Gavrilovića i Marka Đurđevića. Cilj ovog seminarskog rada je pravljenje jednostavne aplikacije na panel-u i njegovo povezivanje na PLC kontroler. Pri ovome su korišćene sledeće hardverske komponente:

- Siemens S7-200(224XP kontroler)
- Siemens Simatic Panel 170micro(Touch Panel u daljem tekstu TP).

Prilikom rada na ovim komponentama bilo je potrebno koristiti specifične softvere koji su inače došli kao njihov deo. Radi se o softverima:

- V4.0 STEP 7 MicroWIN (za programiranje PLC-a)
- SIMATIC WinCC flexible 2004 (za programiranje TP-a).

Dakle krenimo na konkretni rad.

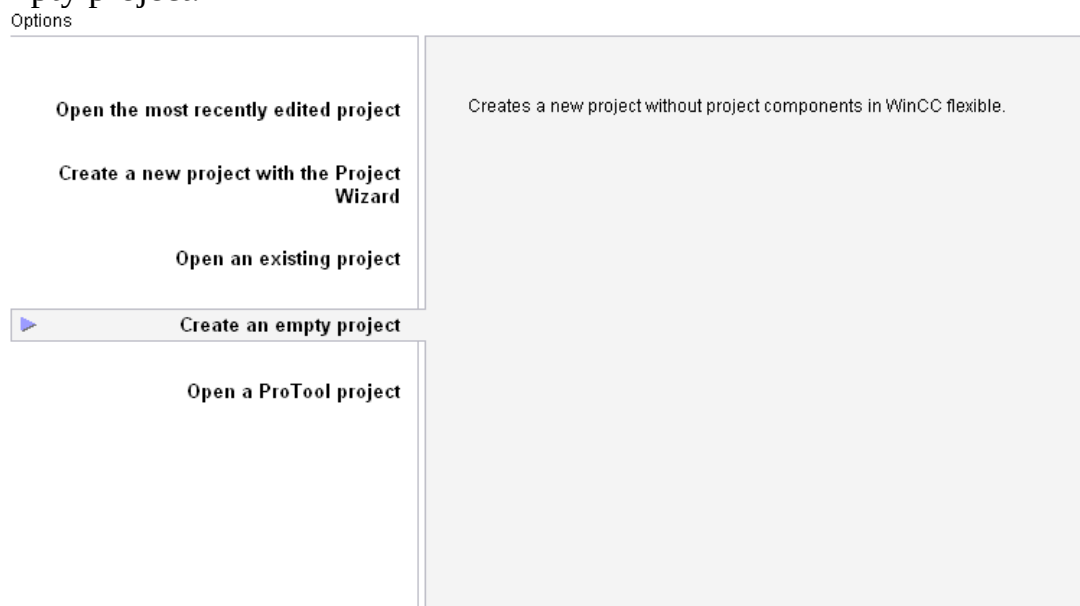
Postavka

Zadatak smo podelili u dve celine:

- Programiranje TP-a
- Programiranje PLC-a.

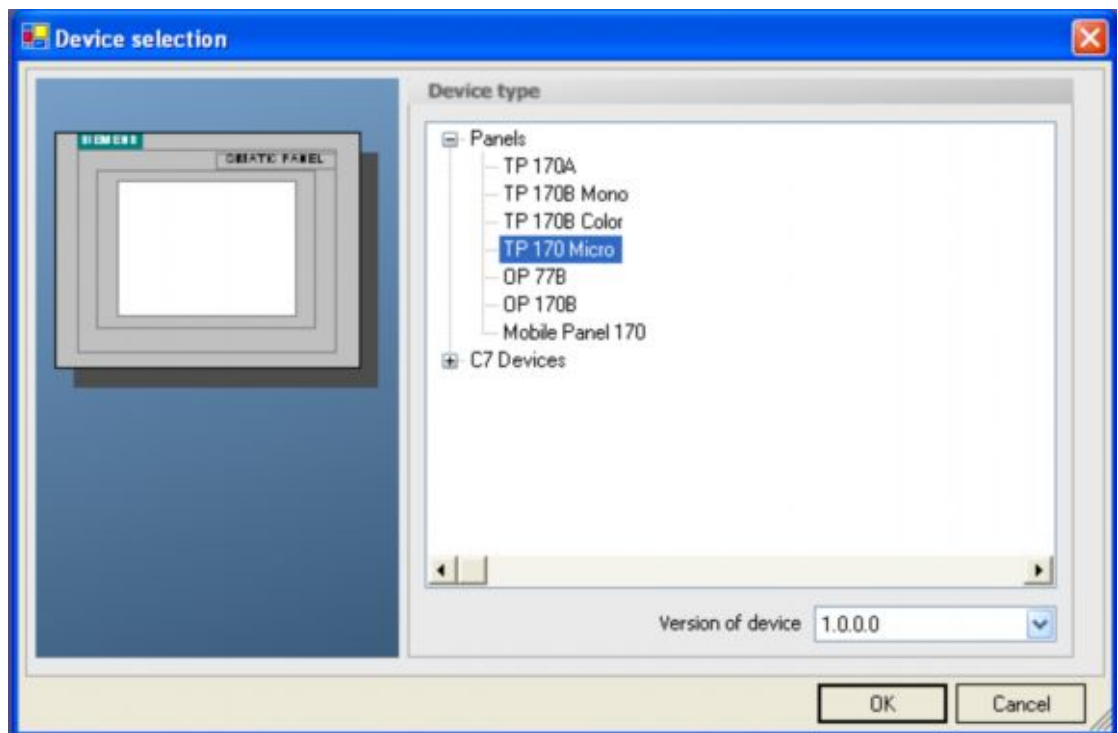
Programiranje TP

Po startovanju programa SIMATIC WinCC flexible 2004 pojavljuje se sledeći prozor u kojem idemo na opciju Create an empty project.



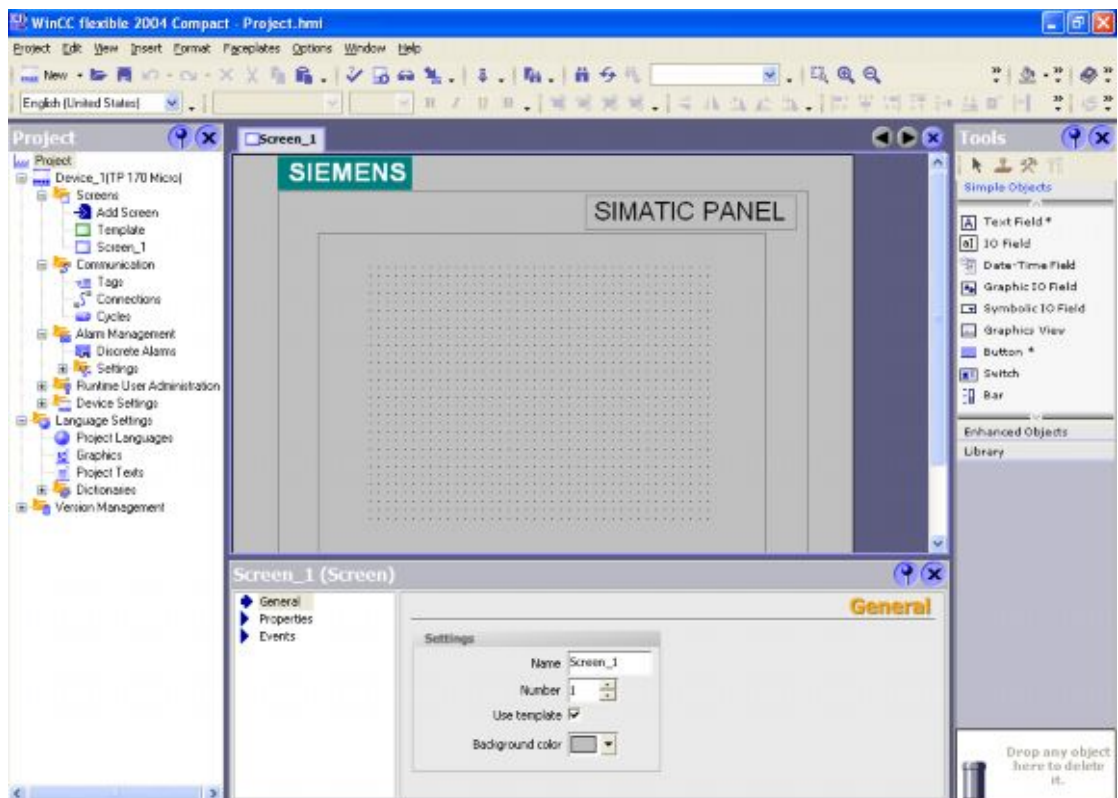
Slika 1.

Nakon ove komande pojavljuje se prozor koji vidimo na narednoj slici u kome treba izabrati tip panela sa kojim radimo a to je u ovom slučaju panel TP 170 Micro.



Slika 2.

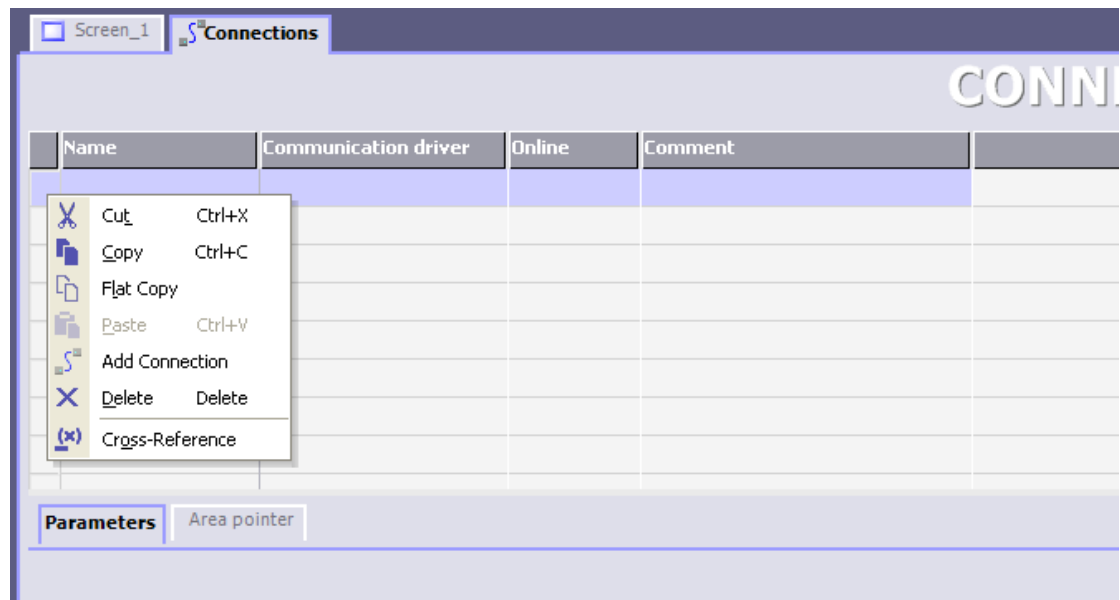
Nakon toga program je otvoren i ima izgled kao na sledećoj slici:



Slika 3.

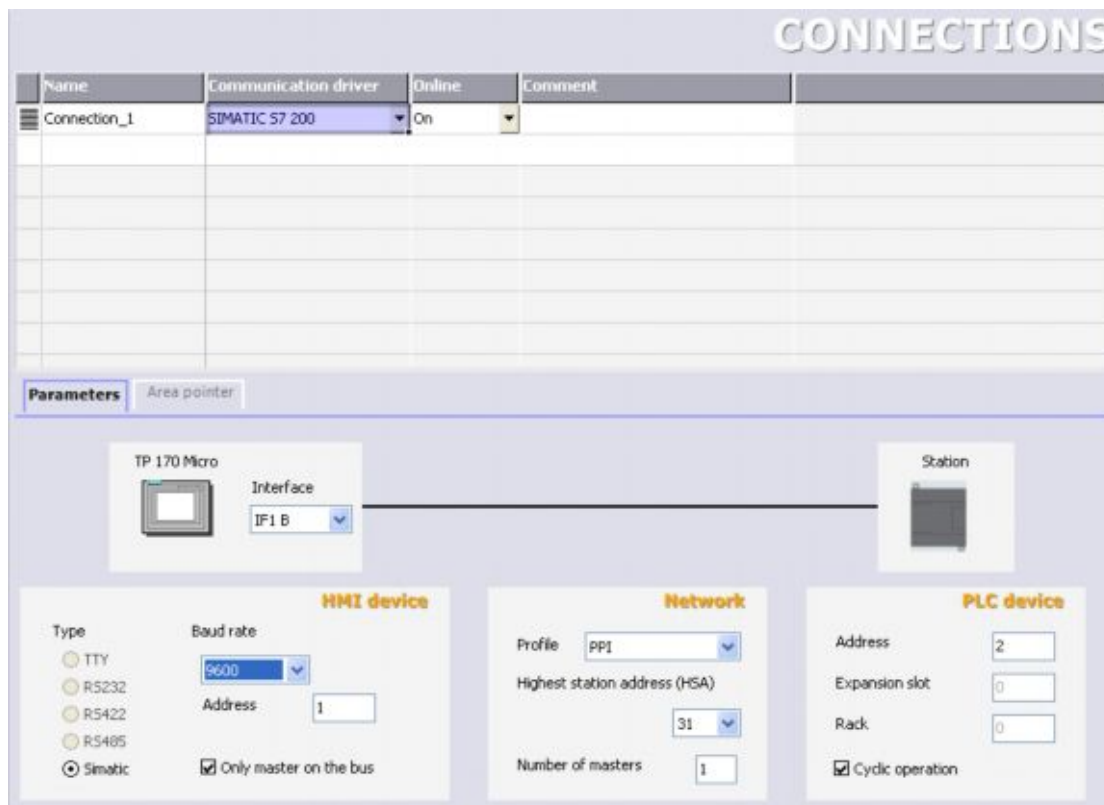
Sa desne strane je *Tools* meni koji se aktivira tako što iz menija *View* kliknemo na opciju *Tools* (ili skraćenica *Ctrl+ Shift+T*). Sada prelazimo na pravljenje aplikacije.

Prvo treba napraviti konekciju tako što idemo na opciju *Connections* koja se nalazi na desnoj strani prozora koji vidimo na slici 3. Pošto kliknemo na *Connections* pojavljuje se prozor koji vidimo na slici 4.



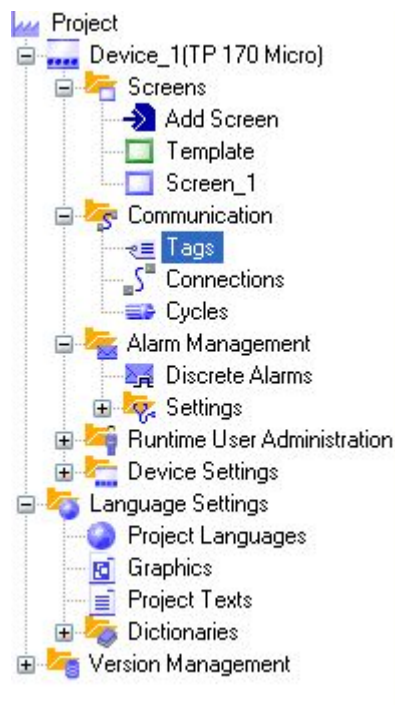
Slika 4.

Idemo na *Add Connection* i pravimo novu konekciju. U polju *Profile* izaberemo *PPI*, a u polju *Baud rate* izaberemo vrednost *9600* što se i može videti na sledećoj slici:



Slika 5.

Sledeće što treba napraviti je *Tag*-ove koji služe deklarisanje grafičkih objekata koji se koriste u aplikaciji . Idemo dvoklikom na komandu *Tags* koja se isto nalazi u meniju sa desne strane i koju možete videti na slici 6.



Slika 6.

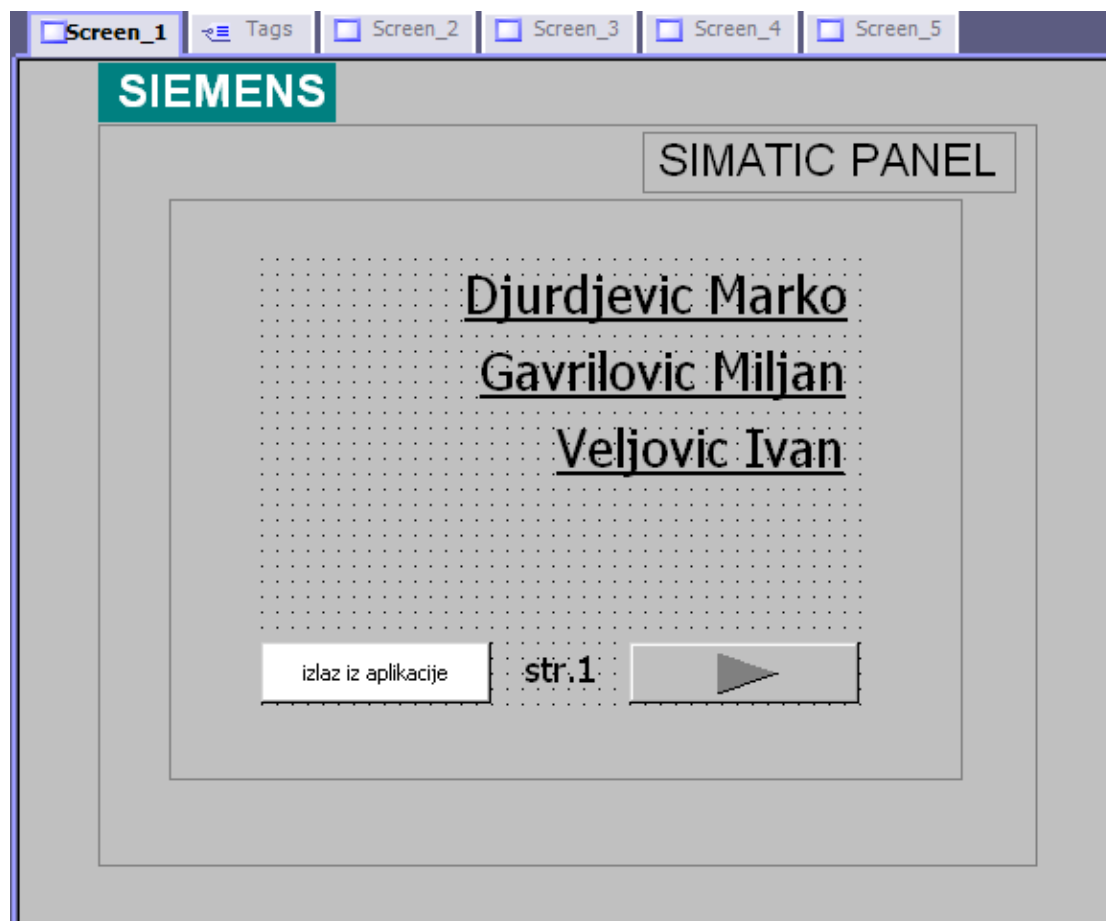
Sada samo treba izvršiti podešavanja koja možete videti na sledećoj slici:

Name ▲	Connection	Data type	Address	Array count	Acquisition cycle
brojac	Connection_1	Int	C 0	1	1 s
motor_01	Connection_1	Bool	M 0.0	1	1 s
motor_02	Connection_1	Bool	M 0.1	1	1 s
signal_03	Connection_1	Bool	M 0.2	1	1 s
signal_04	Connection_1	Bool	M 0.3	1	1 s
signal_05	Connection_1	Bool	M 0.4	1	1 s
signal_06	Connection_1	Bool	M 0.5	1	1 s
signal_07	Connection_1	Bool	M 1.0	1	1 s
signal_08	Connection_1	Bool	M 1.1	1	1 s
signal_09	Connection_1	Bool	M 1.2	1	1 s
signal_10	Connection_1	Bool	M 1.3	1	1 s
signal_11	Connection_1	Bool	M 1.4	1	1 s
signal_12	Connection_1	Bool	M 1.5	1	1 s

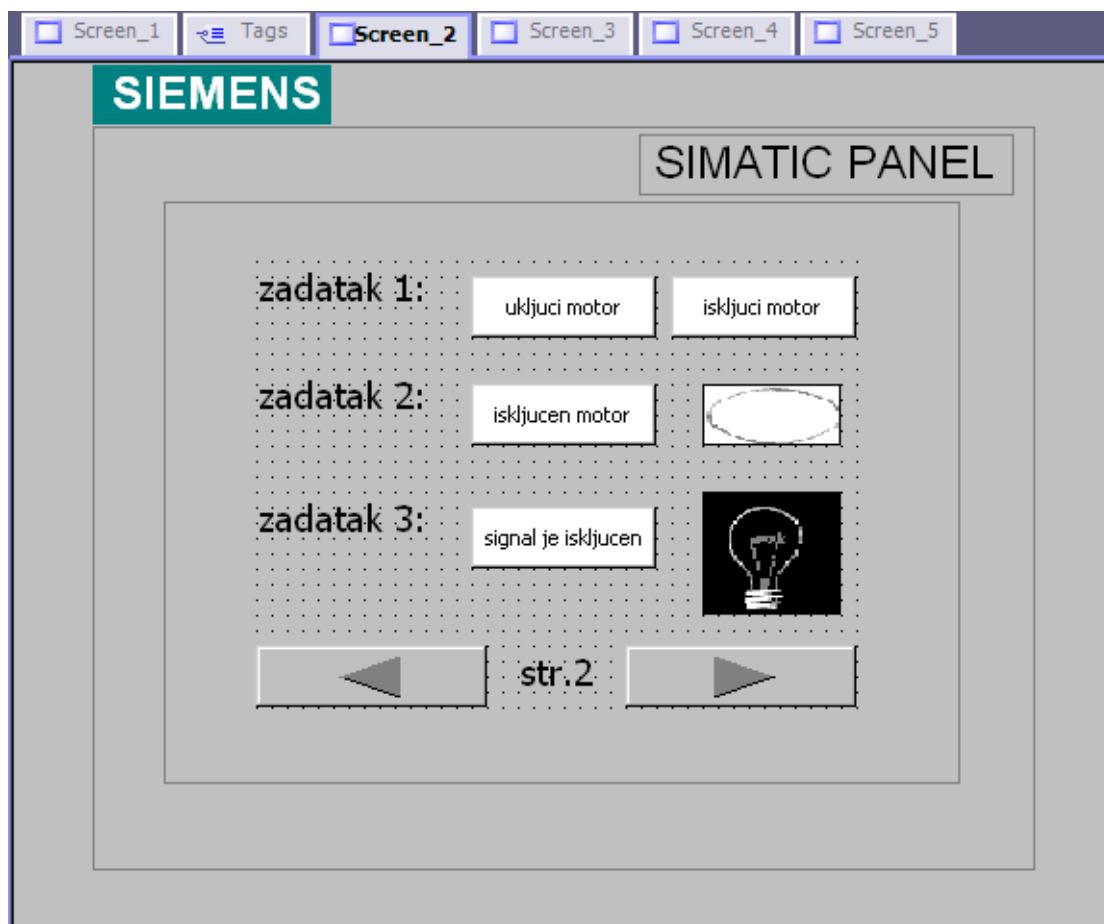
Slika 7.

Kao što možete videti na gornjoj slici za rešavanje naših zadataka mi smo koristili 13 tagova.

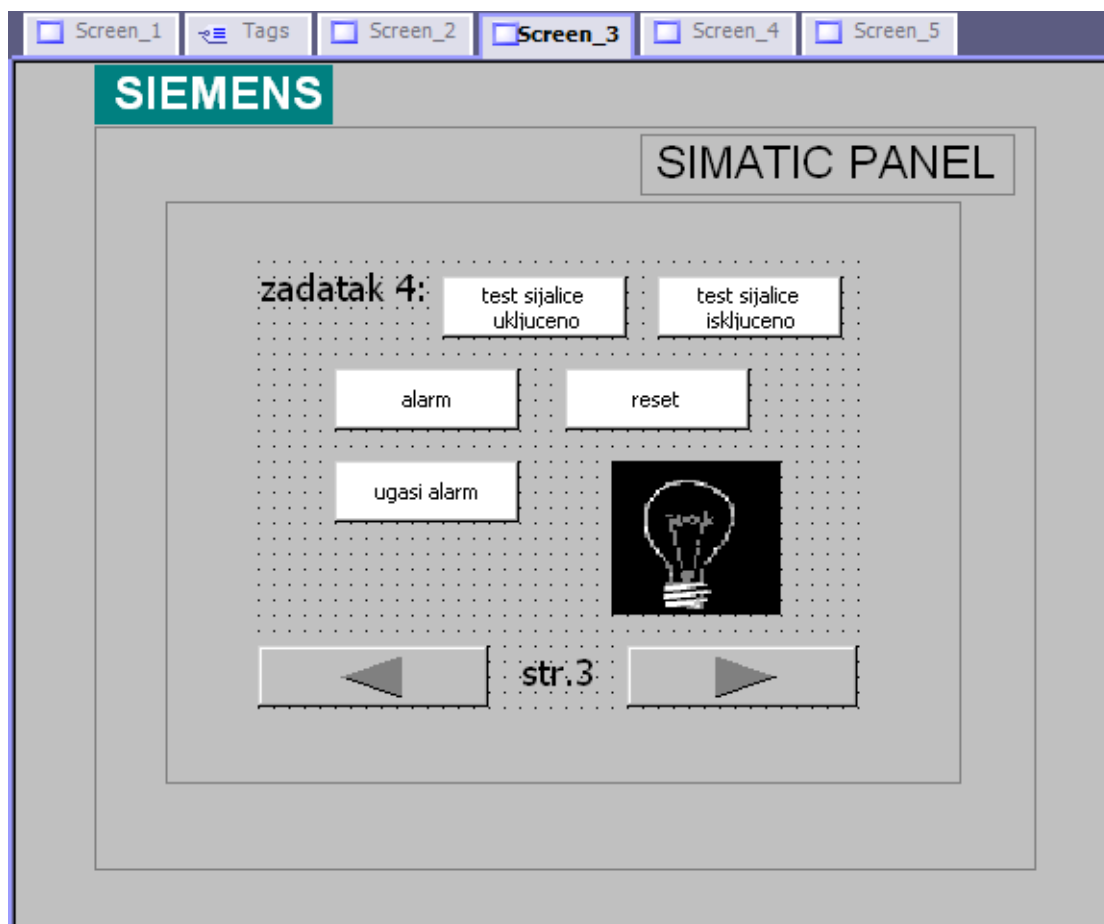
Sledeće što treba napraviti su ekrani (*screens*). Da bi napravili ekran idemo dvoklikom na opciju *Add Screen* koji možete videti na slici 6. Panel koji mi koristimo može da prihvati najviše 5 ekrana koje smo mi i napravili i koje možete da vidite na sledećim slikama:



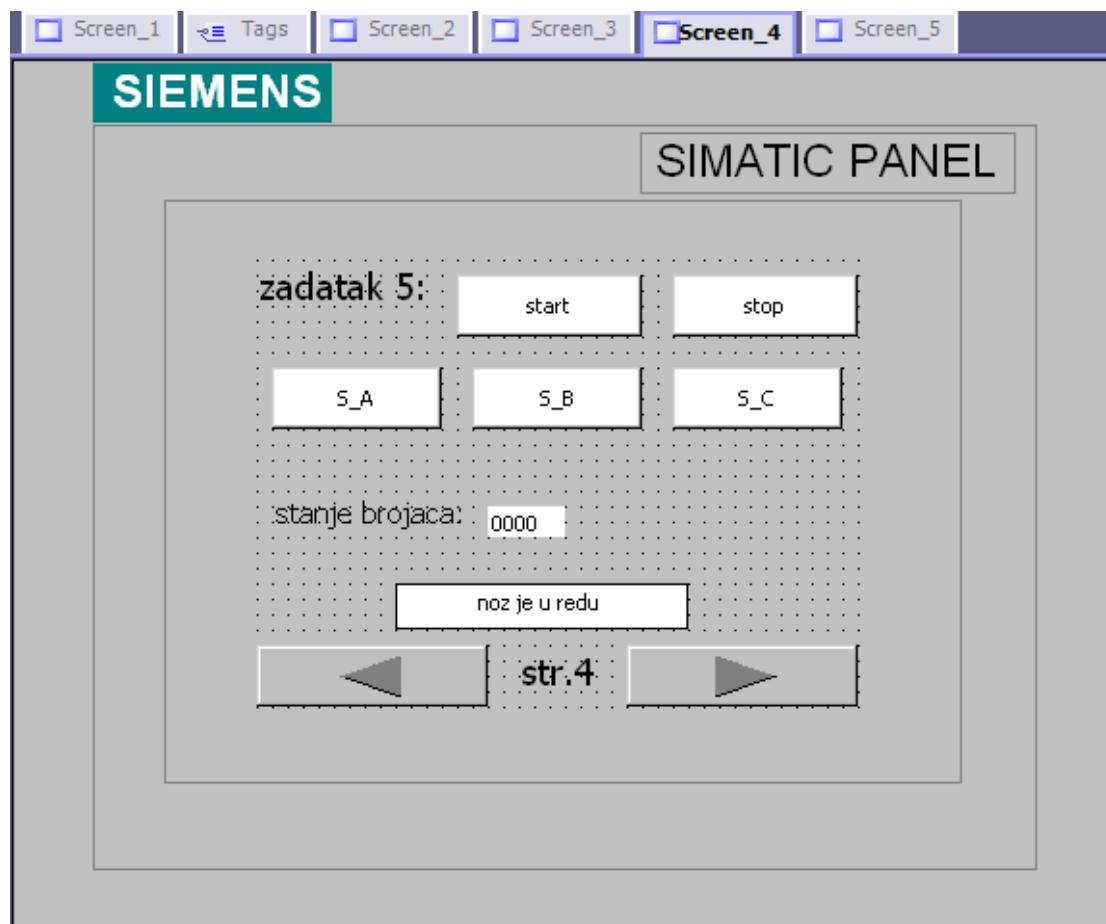
Slika 8.



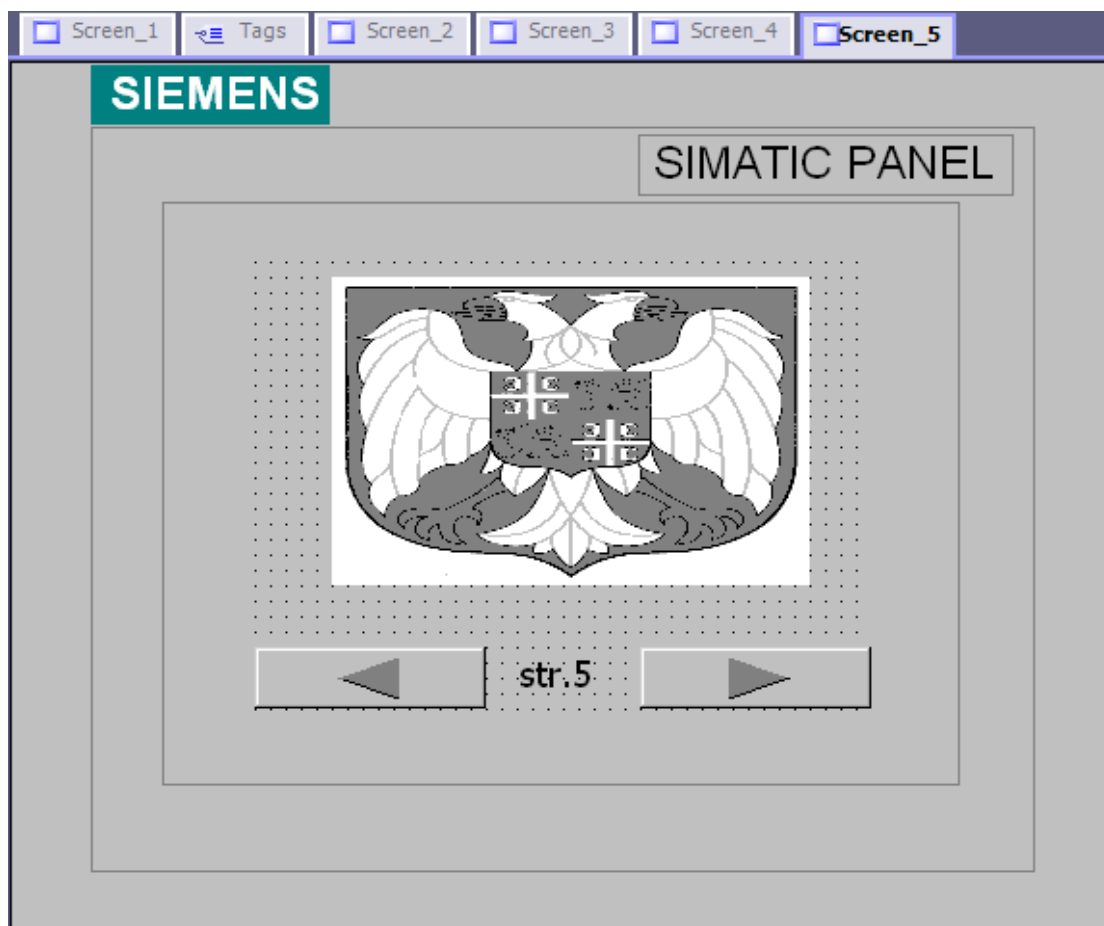
Slika 9.



Slika 10.

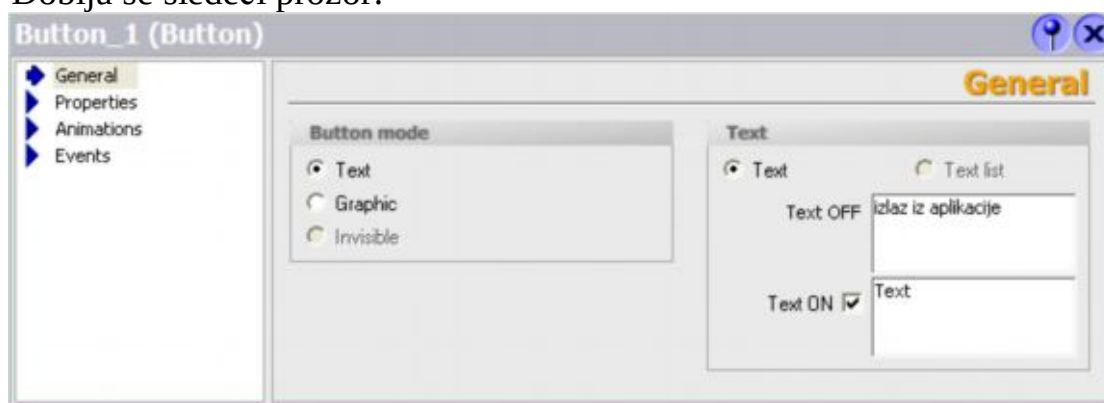


Slika 11.

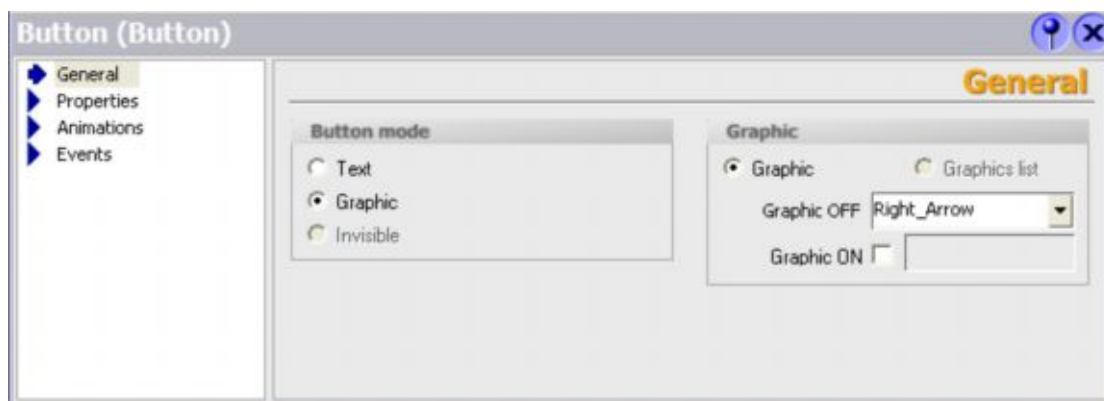


Slika 12.

Kada smo napravili *Screen*-ove potrebno je u njih uneti grafičke objekte. Sada koristimo Tool meni koji se nalazi na desnoj strani prozora što se i može videti na slici 3. Dugme Button se aktivira dvoklikom na njega. Dobija se sledeći prozor:

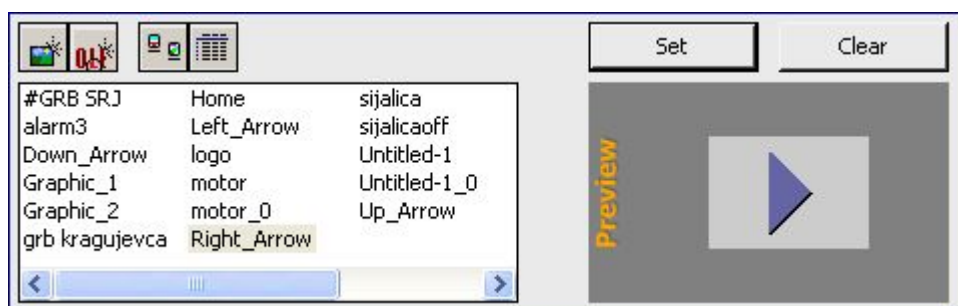


Slika 13.



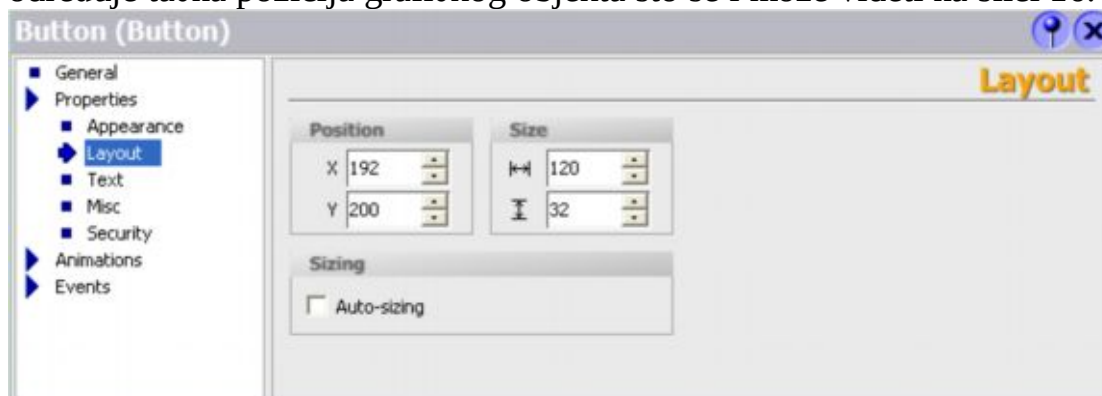
Slika 14.

Sa slika 13 i 14 vidimo da *Button* može da se predstavi i tekstualno i grafički. Da bi ga predstavili grafički idemo na strelicu u čijem polju piše *Right_Arrow* pri čemu se otvara sledeći prozor:



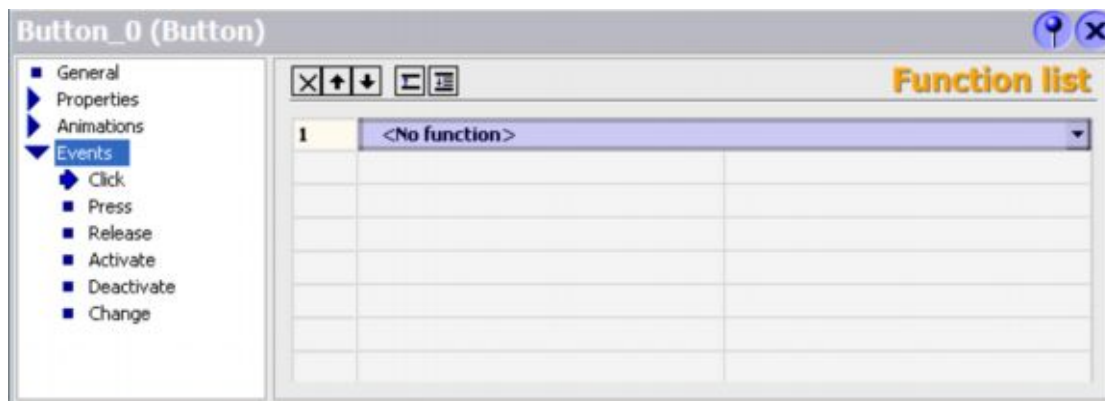
Slika 15.

Oдавde vidimo da sam program sadrži određeni broj slika ali ako nismo zadovoljni sa datim slikama moguće je ubaciti i neku drugu sliku tako što idemo na ikonicu u gornjem levom uglu koja otvara prozor u kojoj biramo lokaciju na kojoj se nalazi koju želimo da ubacimo. Kada izaberemo željenu sliku idemo na komandu *Set*. Kod svakog alata iz menija *Tools* postoji opcija *Layout* pomoću koje se određuje tačna pozicija grafičkog objekta što se i može videti na slici 16.



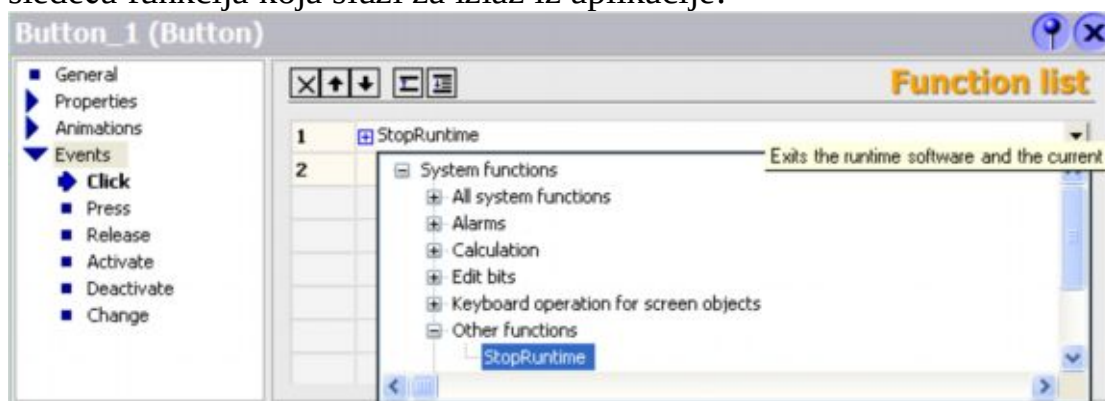
Slika 16.

Sledeći korak je podešavanje funkcije *Button*-a. Idemo na opciju *Events* pa zatim *Click* gde ćemo dobiti prozor sledećeg izgleda:

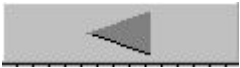


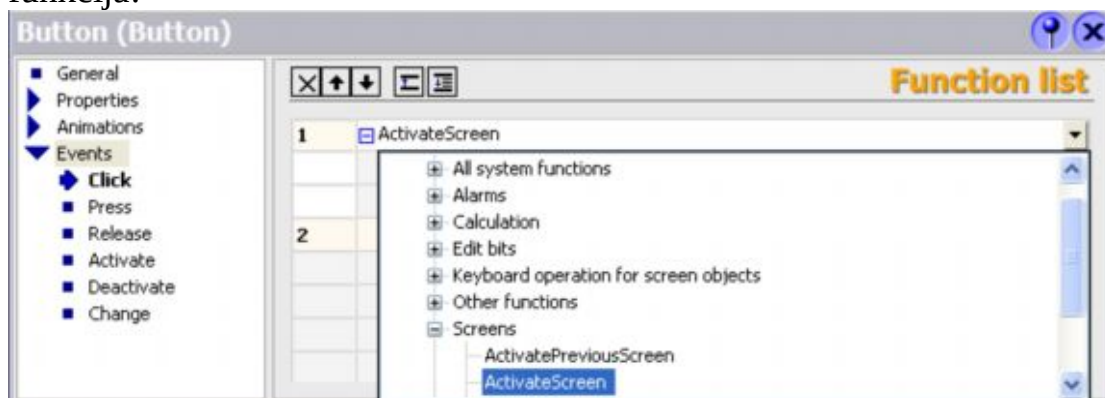
Slika 17.

Klikom na strelicu koju vidimo u gornjem desnom uglu otvara se prozor u kojem bismo željenu funkciju. Za dugme  izabрана je sledeća funkcija koja služi za izlaz iz aplikacije:



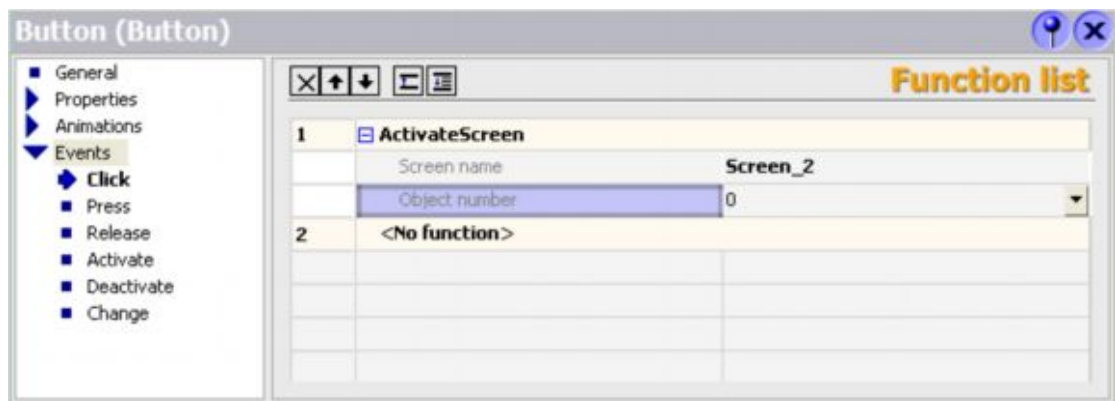
Slika 18.

Za dugmiće  i  birana je sledeća funkcija:



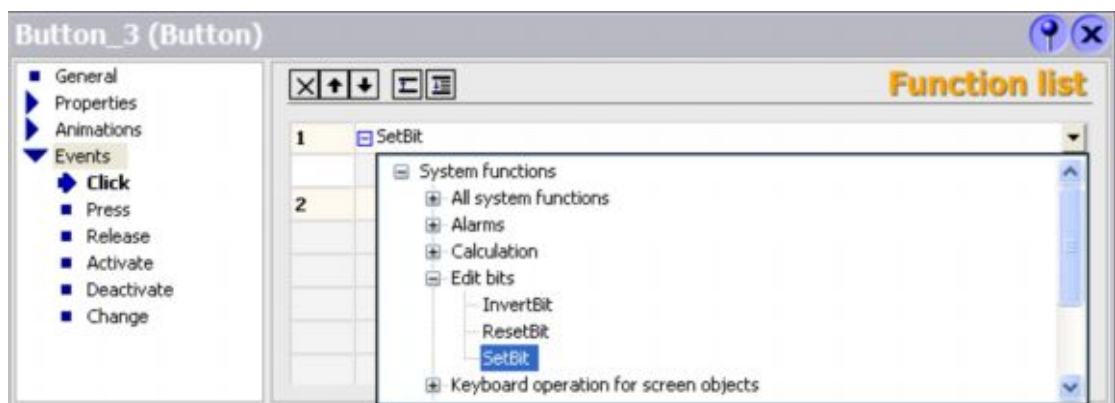
Slika 19.

Ova funkcija služi za prelaz sa jednog ekrana na drugi što znači da u okviru ove funkcije bismo i ekran na koji želimo da nas to dugme prebaci. To možete videti na sledećoj slici:



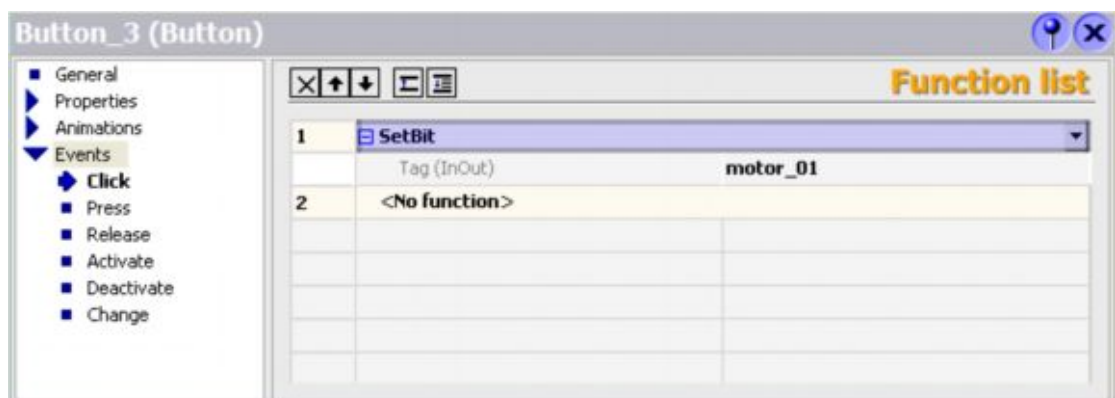
Slika 20.

Sledeća funkcija koju smo koristili pri pravljenju *Button*-a je za setovanje i resetovanje bita na određenoj lokaciji i to možete videti na slici 21:



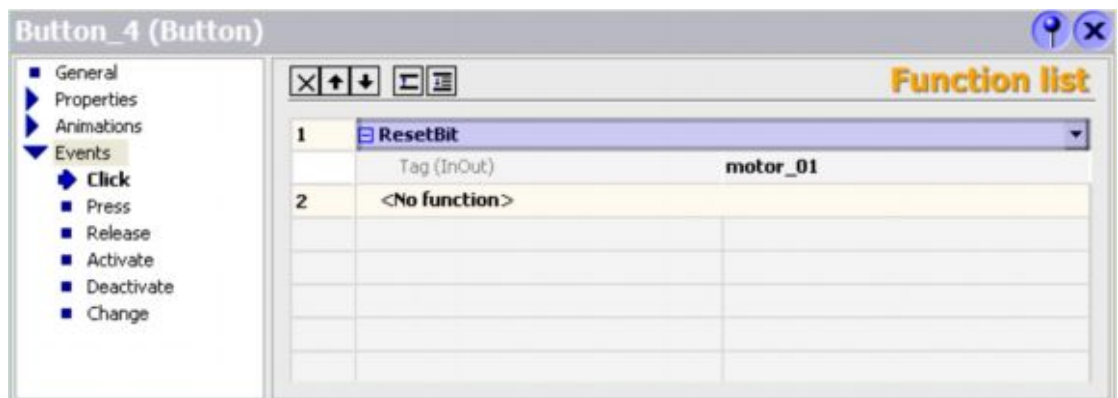
Slika 21.

Naravno u okviru funkcije *SetBit* ili *ResetBit* treba izabrati i *Tag* koji određuje memorijsku lokaciju na kojoj se vrši funkcija. Tako smo za dugme ukljuci motor koristili funkciju *SetBit* što se vidi na slici 22:

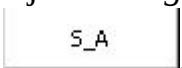
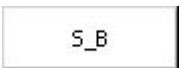


Slika 22.

Za dugme  je korišćena funkcija *ResetBit* sa istim *Tag*-om što se može videti na sledećoj slici:



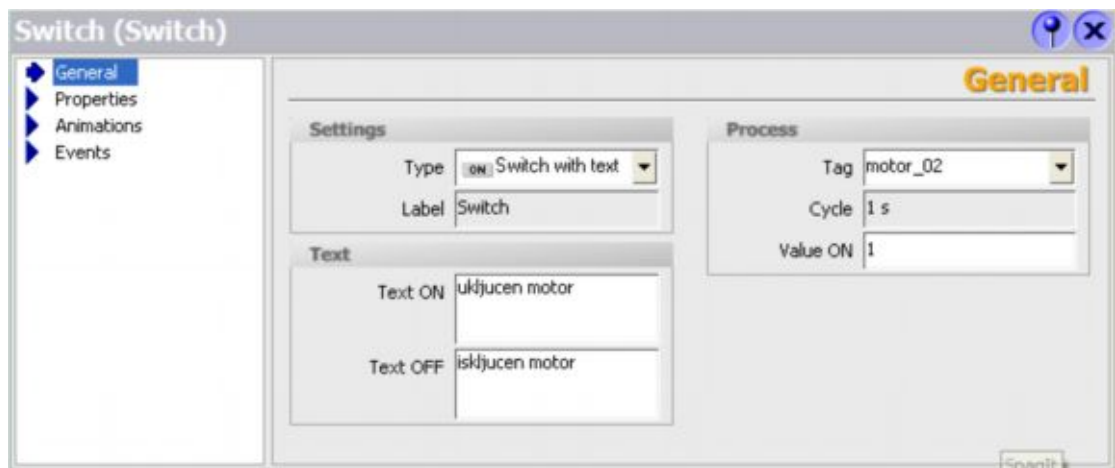
Slika 23.

Takođe jedno dugme može da sadrži i dve funkcije odjednom (misli se na *SetBit* i *ResetBit* koji naravno imaju isti *Tag*). To smo koristili kod sledećih dugmića: , ,  i . Dakle pritiskom na ovo dugme setuje se bit na određenoj lokaciji za određeni vremenski interval a zatim se taj bit resetuje. Kako su funkcije odrađene možete videti na sledećoj slici:



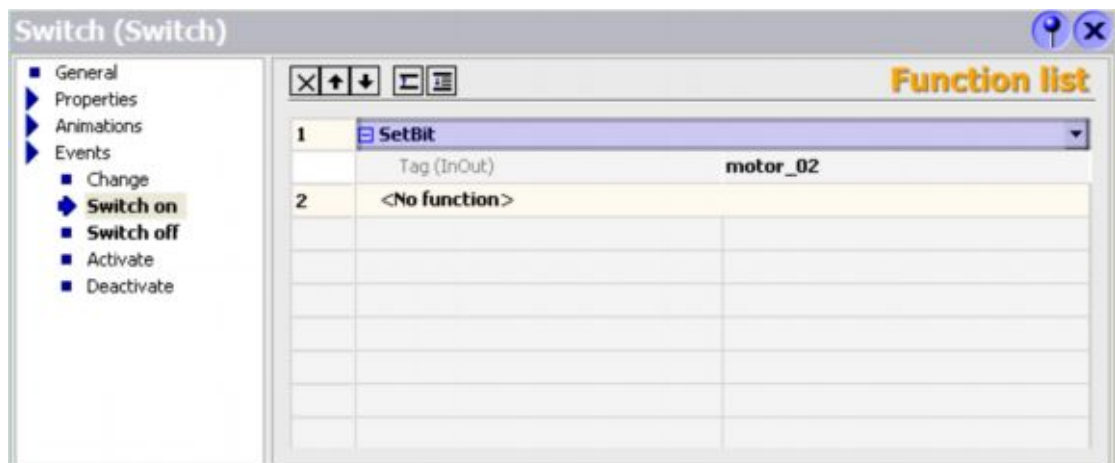
Slika 24.

Sledeći alat iz *Tools* menija koji smo koristili je *Switch* i čiji prozor možemo videti na slici 25:

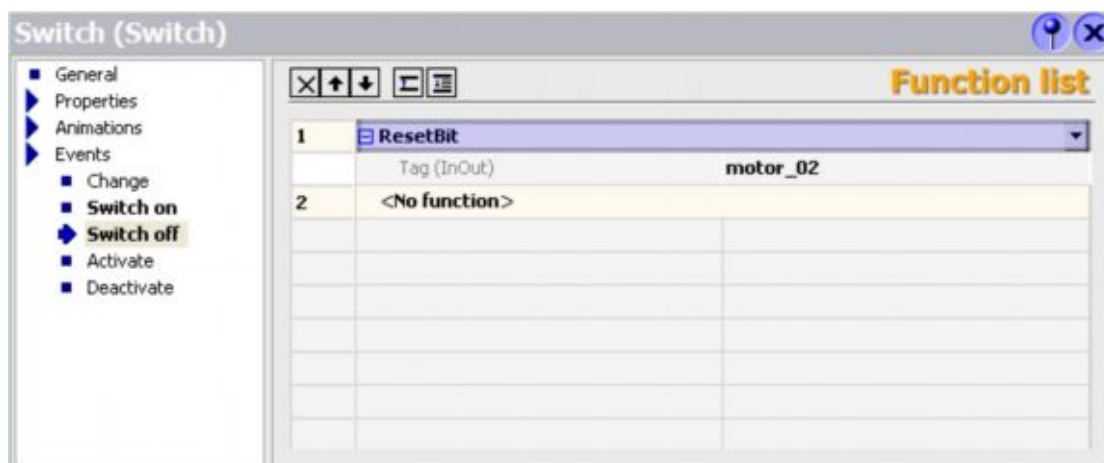


Slika 25.

Kao što se sa slike može videti u gornjem desnom uglu biramo *Tag* za ovu funkciju. U donjem levom uglu upisujemo tekst koji će pisati na dugmetu kada je uključeno (*Text ON*) i kada je isključeno (*Text OFF*). Sa slika 26 i 27 vidimo da u opciji *Events* imamo *Switch on* i *Switch off*. U opciji *Switch on* biramo funkciju *SetBit* a u opciji *Switch off* biramo funkciju *ResetBit*.

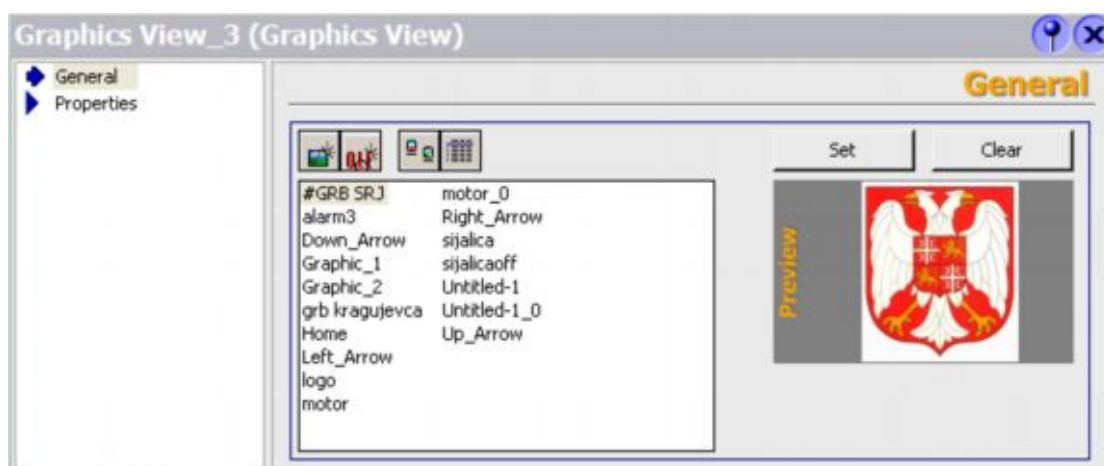


Slika 26.



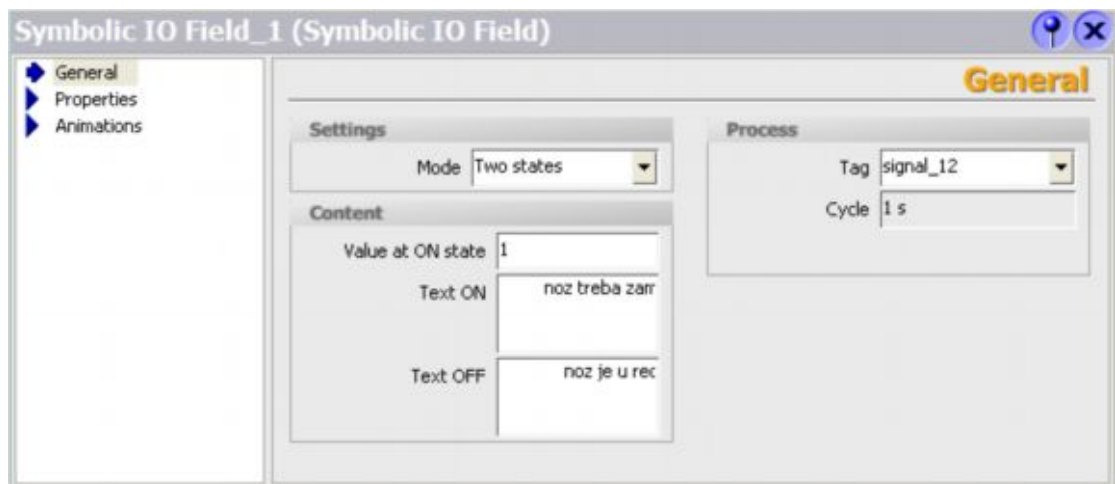
Slika 27.

Alat *Graphics View* služi samo za unošenje slika tj. uloga mu je da ulepša sam izgled panela. Princip unošenja slika je isti kao i kod grafičkog *Button*-a koji je već objašnjen. Izgled prozora ovog alata je prikazan na sledećoj slici:



Slika 28.

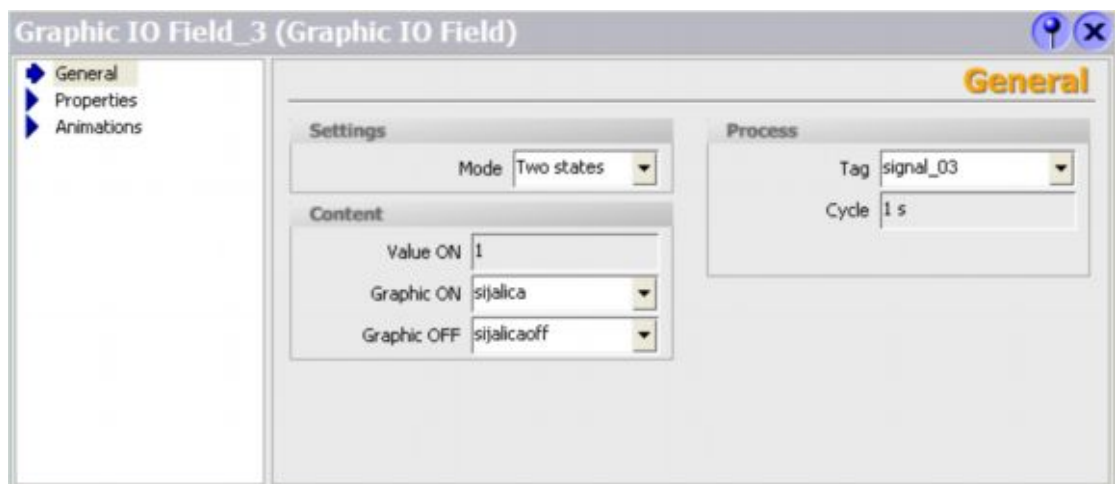
Symbolic IO Field je alat koji služi za tekstualnu vizealizaciju. Kao što možemo videti na slici 29. *Symbolic IO Field* ima dva stanja (kada prima i kada ne prima signal).



Slika 29.

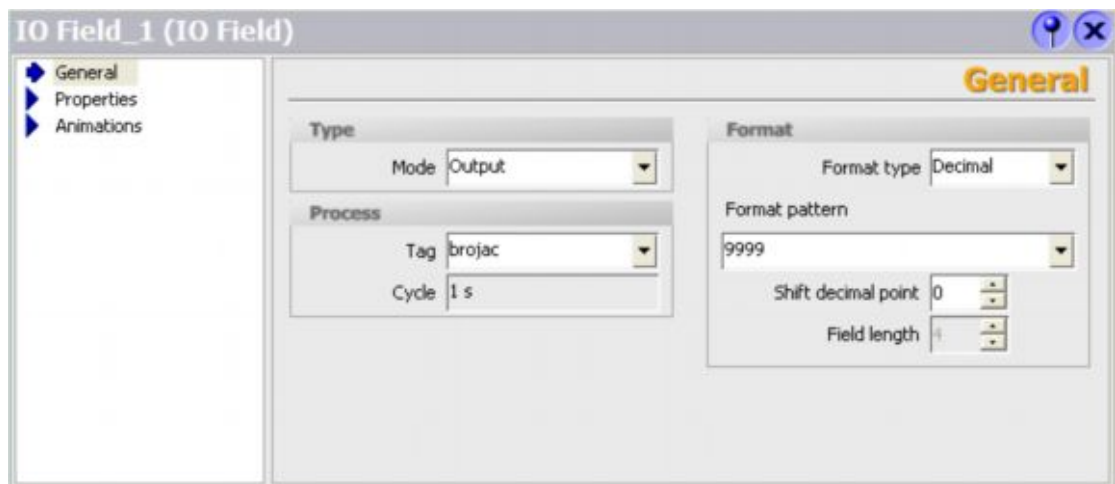
Kao što možemo videti sa slike u gornjem desnom uglu biramo određeni *Tag* dok u donjem desnom uglu unosimo tekst. Tekst u polju *Text OFF* je aktivan sve dok *Symbolic IO Field* ne počne da prima signal kada se aktivira tekst u polju *Text ON*.

Graphic IO Field je u principu isti kao i *Symbolic IO Field*. Sa slike 30. možemo videti da su podešavanja ista kao i kod *Symbolic IO Field* jedino je razlika u tome što se ovde ne unosi tekst već slike. Unos slika je isti kao i kod grafičkog *Button*-a.



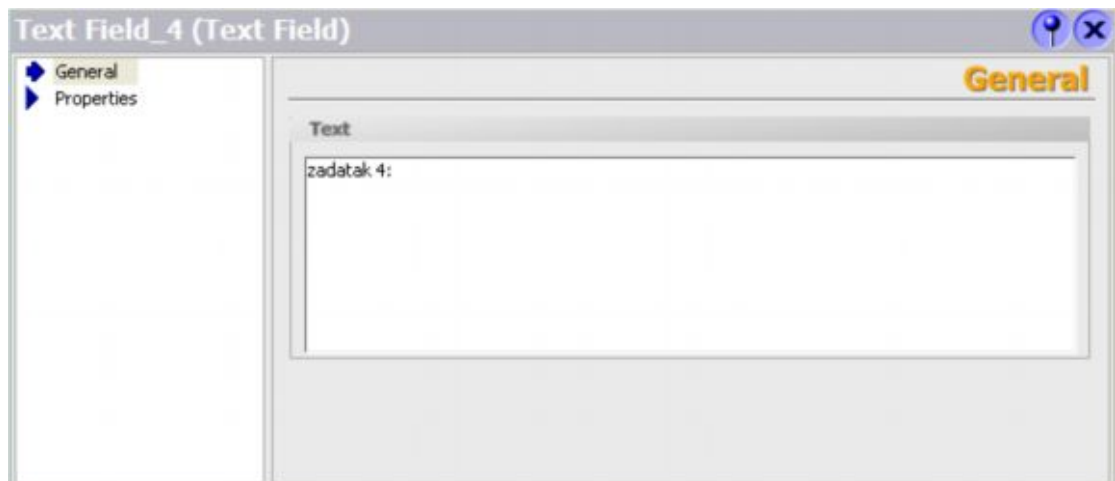
Slika 30.

Alat *IO Field* prikazan je na slici 31. Kao što se može videti sa slike u donjem levom uglu biramo odgovarajući *Tag*. U gornjem levom uglu se nalazi polje *Type* gde biramo kakav mod želimo (input, output ili input/ output). U gornjem desnom uglu je polje *Format pattern* u kojem biramo broj cifara koje trebaju biti prikazane u polju 0000.



Slika 31.

Text Field služi samo unošenje teksta na panel. Njegov prozor možemo videti na slici 32.

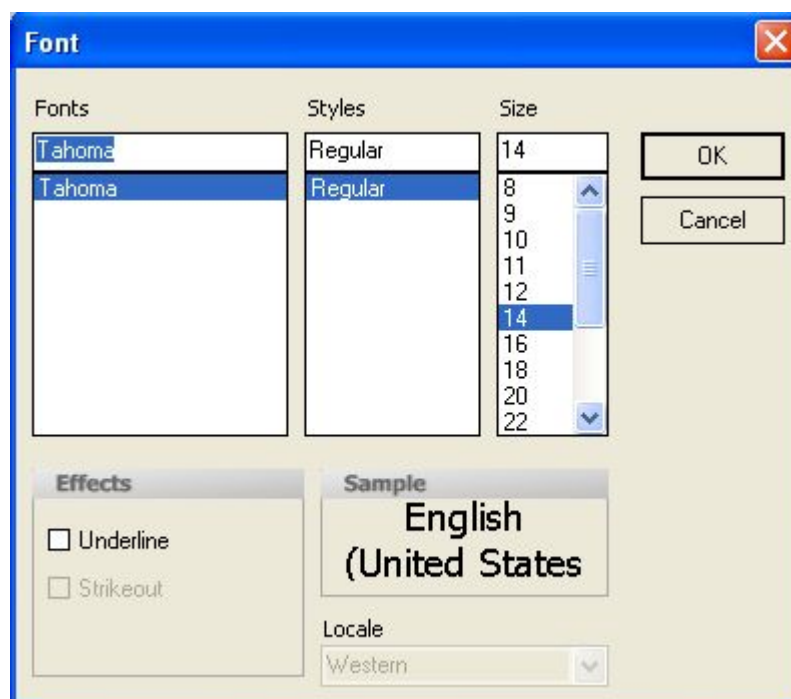


Slika 32.

U *Properties*-u u opciji *Text* možemo birati veličinu teksta kao i da li želimo da je tekst podvučen ili ne. To ćemo uraditi tako što kliknemo na polje *Font* pri čemu se otvara prozor na slici 33. odakle bismo veličinu teksta.



Slika 32.



Slika 33.

Pošto smo završili sa pravljenjem aplikacije potrebno je proveriti da li aplikacije radi. Klikom na ikonu  vrši se kompajliranje cele aplikacije. Posle provere ostaje nam još da prebacimo aplikaciju sa računara na panel. Za prebacivanje podataka koristimo R232/PPI Multi-Master kabl. U WinCC-u kliknemo na ikoncu za transfer koja izgleda ovako  čime će naša aplikacija biti prebačena na TP. Neophodno je još isprogramirati PLC i povezati na TP čime smo spremni za prvi test naše aplikacije.

Programiranje PLC-a

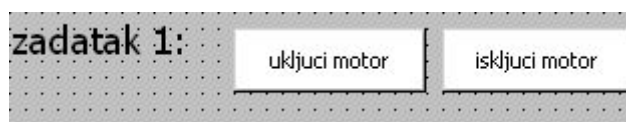
Programiranje PLC-a se vrši u programu V4.0 STEP 7 MicroWIN. Videćete osam zadataka urađenih u PLC-u.

Zadatak 1

Napisati LAD program za PLC, kojim se na pritisak tastera START, startuje motor, a na pritisak tastera STOP isti motor zaustavlja.

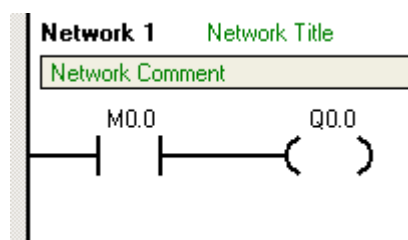
Rešenje

Za upravljanje ovim zadatkom tj. aplikacijom koristimo sledeće komande sa TP-a:



Slika 34.

Ova dva dugmeta setuju i resetuju bit M0.0 u memoriji PLC-a. Na osnovu ovog bita pravimo aplikaciju za PLC datoj na sledećoj slici:



Slika 35.

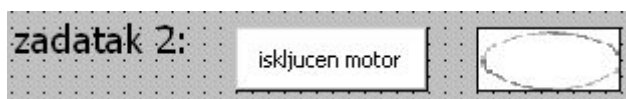
Sa slike se jasno vidi da će u zavisnosti od bita M0.0 biti setovan i izlaz Q0.0.

Zadatak 2

Napisati porogram za PLC, kojim se motor pokreće pritiskom na taster,i pritiskom na isti taster zaustavlja.

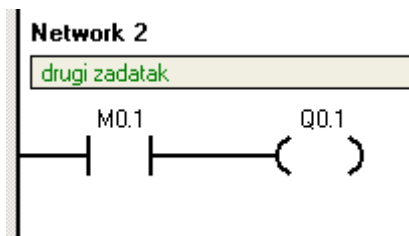
Rešenje

Komande kojima se upravlja ovom aplikacijom možemo videti na sledećoj slici:



Slika 36.

U ovom primeru smo koristili jedno switch dugme. Kada je ono pritisnuto setuje se bit na lokaciji M0.1 i dugme ima 3-D efekat. Kada se opet pritisne dugme bit se resetuje. Na osnovu ovog bita pravimo aplikaciju za PLC datoj na sledećoj slici:



Slika 37.

Sa slike se kao i prethodnom primeru vidi da je stanje na izlazu određeno stanjem bita M0.1 u memoriji PLC-a.

Zadatak 3

Napisati PLC program kojim će se obezbediti kašnjenje ulaznog signala od 2.5 sekunde na izlazu Q0.2. Izlazni signal nastaje 5 sekundi posle početka ulaznog signala i koji će delovati još 7 sekundi po presktanku ulaznog signala na izlazu Q0.3. Izlazni signal na Q0.4 se ponavlja u određenim vremenskim intervalima, kada se na ulaz dovodi signal. Na izlazu Q0.5 se pravi duže kašnjenje signala na izlazu od maksimalnog vremena kašnjenja jednog tajmera kada se ulazni signal iskljuci.

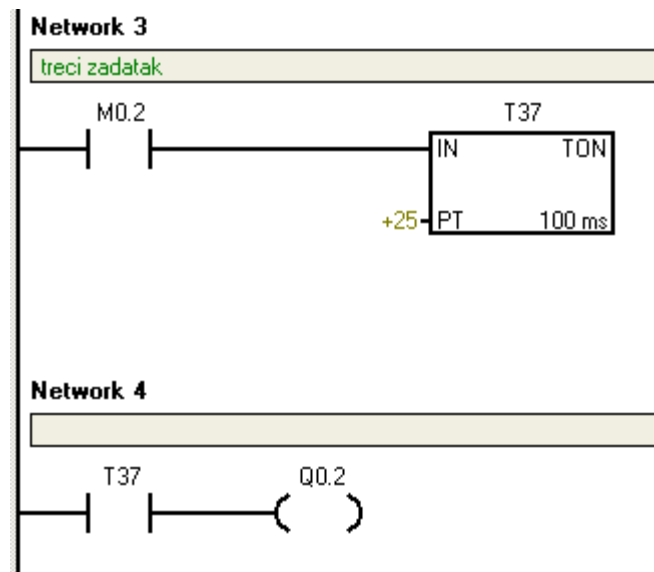
Rešenje

Komande za upravljanje u PLC-u su sledeće:



Slika 38.

U ovom primeru smo koristili jedno switch dugme. Kada je ono pritisnuto setuje se bit na lokaciji M0.2 i dugme ima 3-D efekat. Kada se opet pritisne dugme bit se resetuje. Na osnovu ovog bita pravimo aplikaciju za PLC datoj na sledećoj slici:



Slika 39.

Sa slike se jasno vidi da nakon setovanja bita M0.2 tajmer (T37 je adresa tajmera) počinje da odbrojava i kada dostigne vrednost od 2.5s aktivira se njegov izlaz TON što automatski aktivira izlaz Q0.2.

Zadatak 4

Napisati LAD program kojim izlazni signal nastaje 5 sekundi posle početka ulaznog signala i koji će delovati još 7 sekundi po presktanku ulaznog signala.

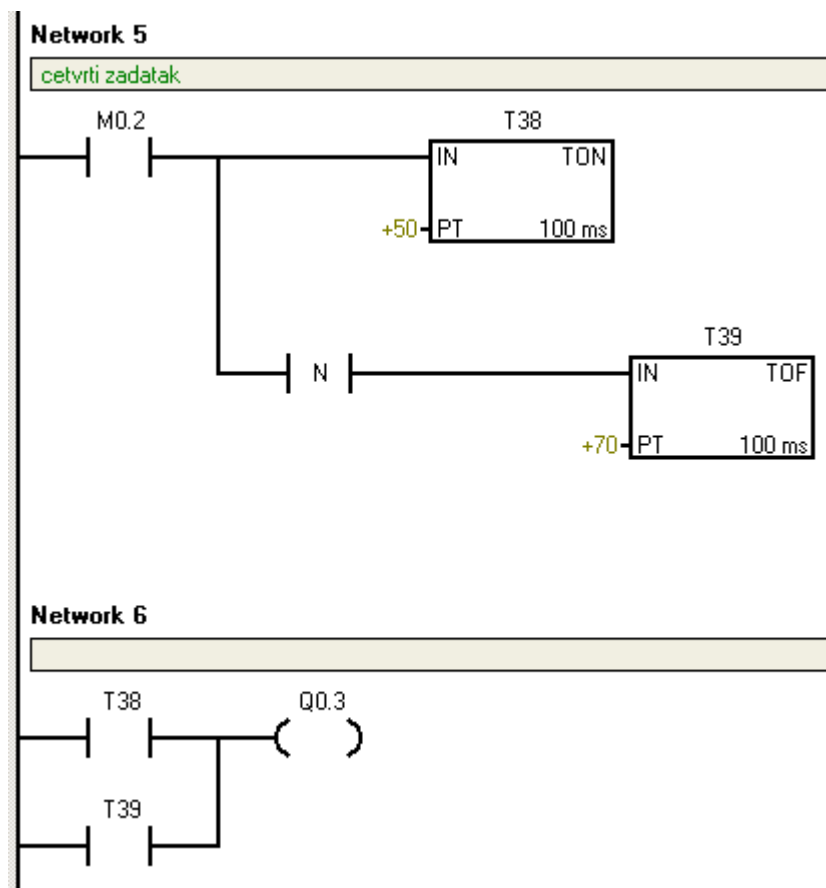
Rešenje

Komande za upravljanje na TP-u su iste kao i u prethodnom zadatku:



Slika 40.

U ovom primeru koriste se 2 tajmera (T38 i T39) kao na sledećej slici:



Slika 41.

Tajmer T38 ima zadatak da napravi kašnjenje od 5s, nakon setovanja bita M0.2 dok tajmer T39 pravi kašnjenje od 7s nakon resetovanje bita M0.2. Na taj način se dolazi do rešenja problema u zadatku.

Zadatak 5

Napisati LAD program koji ponavlja izlazne signale u određenim vremenskim intervalima, kada se na ulaz dovodi signal.

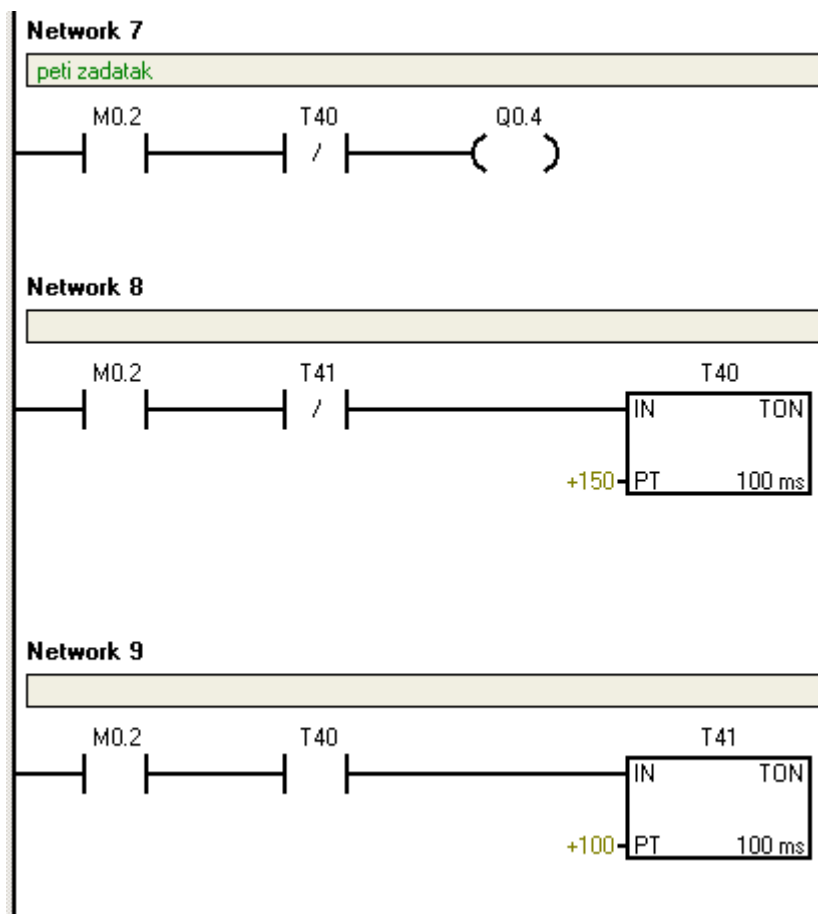
Rešenje

I u ovom zadatku koristimo iste komande kao u prethodna zadatka i koje možete videti na sledećoj slici:



Slika 42.

Za rešavanje ovog zadatka koristimo bit sa lokacije M0.2 i dva tajmera (T40 i T41) kao na sledećoj slici:



Slika 43.

Sa slike se vidi da se nakon setovanja bita M0.2 uključuje tajmer T40 koji aktivira izlaz Q0.4. Isključenjem tajmera T40 aktivira se T41 koji određuje vreme resetovanje izlaza Q0.4.

Zadatak 6

Napisati LAD program koji omogućava duže kašnjenje signala na izlazu od maksimalnog vremena kašnjenja jednog tajmera.

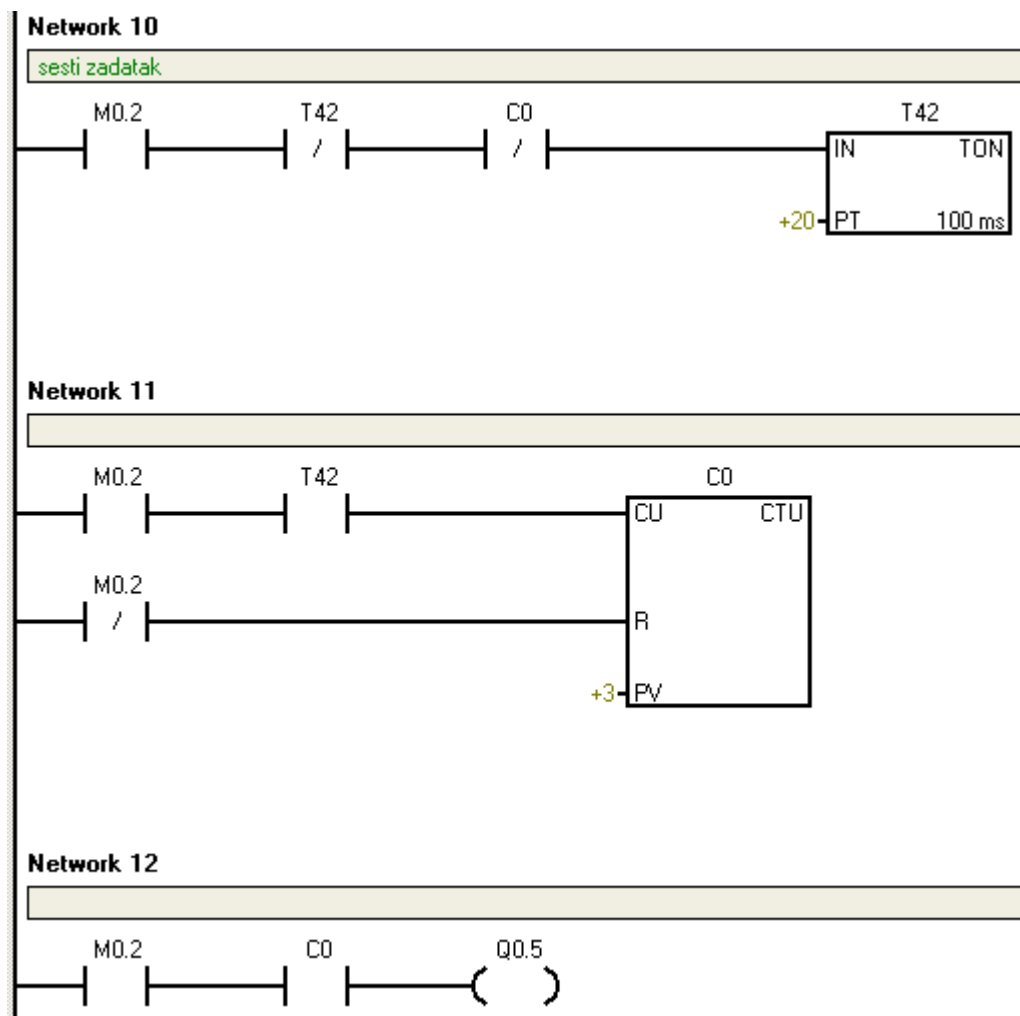
Rešenje

Na sledećoj slici je prikazan izgled aplikacije za upravljanje na TP-u:



Slika 44.

Za rešavanje ovog zadatka koristimo bit sa lokacije M0.2, tajmer T42 i brojač C0 kao što je prikazano na sledećoj slici:



Slika 45.

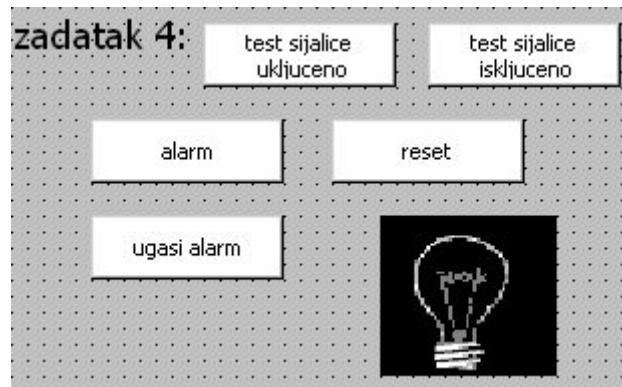
Nakon uključivanja ulaza, tajmer T42 i brojač C0 su isključeni, pali se T42 (*Network 1*), i odbrojava predefinisano vreme kašnjenja koje množimo sa brojem koji zadajemo brojaču C0 (u ovom primeru $20 \times 100\text{ms} = 2\text{s}$). Nakon isteka prvog perioda tajmera, pali se njegov izlaz i trenutna vrednost brojača se povećava za jedan (*Network 2*), ali se u isto vreme ulaz tajmera gasi, tajmer se resetuje i ponovo počinje odbrojavanje (*Network 1*).

Zadatak 7

Kao posledica uključenja alarma, pali se alarmna sijalica (signal sa prekidima – treptanje) i alarmna sirena (kontinualan signal). Pritiskom na reset taster (konstatovano je da postoji kvar), alarmna sirena se isključuje, a alarmna sijalica radi kontinualno dok se ne otkloni uzrok alarma. Napisati LAD program koji na način prikazan na vremenskom dijagramu pali i gasi alarmno svetlo i sirenu.

Rešenje

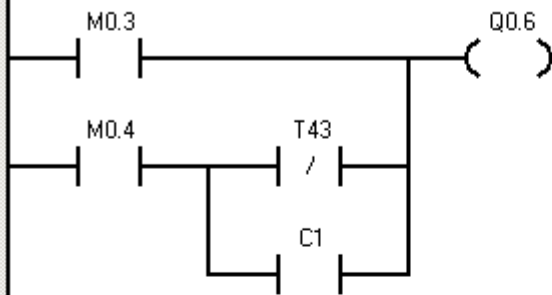
Za upravljanje u ovom zadatku koristi se sledeća aplikacija na TP-u:



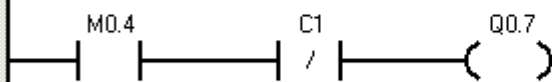
Slika 46.

Network 13

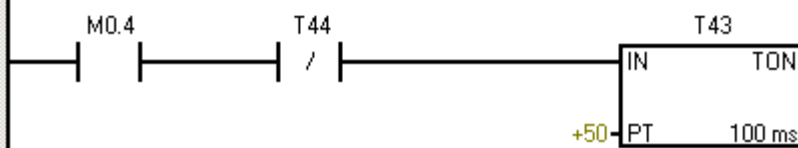
sedmi zadatak



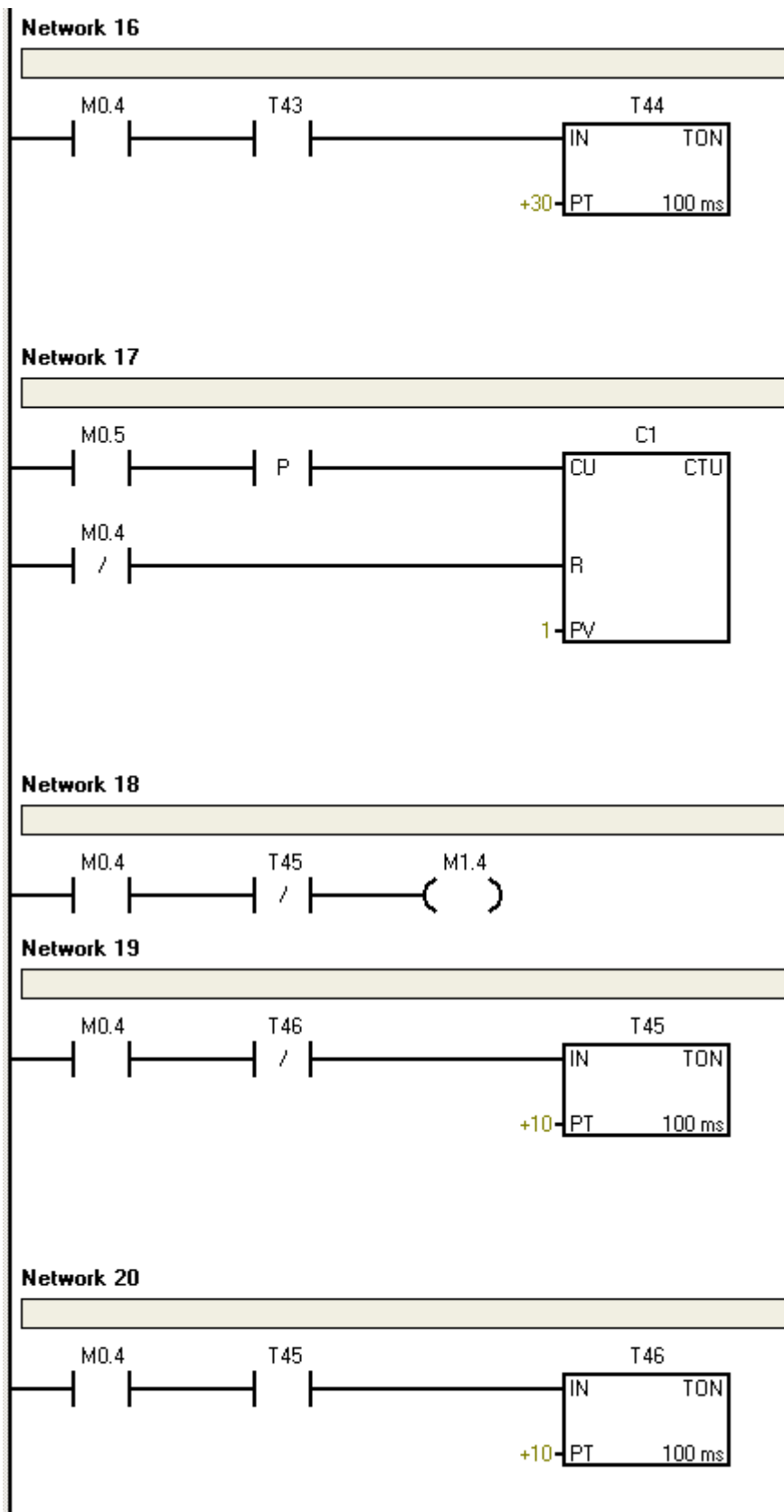
Network 14



Network 15



Slika 47.



Slika 48.

Prvo proveravamo ispravnost test sijalice. Pritiskom na dugme Test_sijalice uključeno setuje se bit na lokaciji M0.3 i pali si sijalica Q0.6, a pritiskom na dugme Test_sijalice isključeno gasi se sijalica.

Stanje tajmera i brojača je nula (OFF). Usled nastanka alarma M0.4, pali se sijalica Q0.6 (*Network 1*) i alarmna sirena Q0.7 (*Network 2*). Pali se tajmer T43 (*Network 3*), i odbrojava predefinisanu vrednost koja određuje dužinu trajanja uključenog stanja sijalice Q0.6 (u ovom slučaju: $50 \times 100 \text{ ms} = 5 \text{ s}$).

Nakon 5 sekundi, tajmer T43 pali izlaz TON (OFF u ON), gasi se sijalica (*Network 1*), startuje se tajmer T44 (*Network 4*), i odbrojava predefinisanu vrednost koja određuje dužinu trajanja isključenog stanja sijalice (u ovom slučaju: $30 \times 100 \text{ ms} = 3 \text{ s}$).

Nakon 3 sekunde, sijalica je ugašena, tajmer T44 pali svoj izlaz (*Network 4*), tajmer T43 gasi svoj izlaz, vrednost njegovog brojača se resetuje (*Network 3*), tajmer T44 gasi svoj izlaz i takođe resetuje vrednost svog brojača (*Network 4*), i sijalica se ponovo pali (*Network 1*).

Ovaj proces se nastavlja do trenutka prestanka postojanja alarma ili do trenutka pritiska RESET tastera. Sve vreme sirena Q0.7 ostvaruje kontinualni rad. Odmah po pritisku tastera RESET, startuje se brojač C1 koji povećava svoju trenutnu vrednost na 1.

Takođe da bismo omogućili treperenje slike u *Graphic IO Field*-u na TP-u dodata su još tri nova reda (*Network 18*, *19* i *20*). Možete videti da koristimo još dva tajmera T45 i T46. Princip rada je isti kao i sa tajmerima T43 i T44 samo što predefinisana vrednost koja određuje dužinu trajanja uključenog tj. isključenog stanja *Graphic IO Field*-a u ovom slučaju $10 \times 100 \text{ ms} = 1 \text{ s}$. Izlaz je stavljen na memorijsku lokaciju M1.4 na koju je vezan i *Graphic IO Field*.

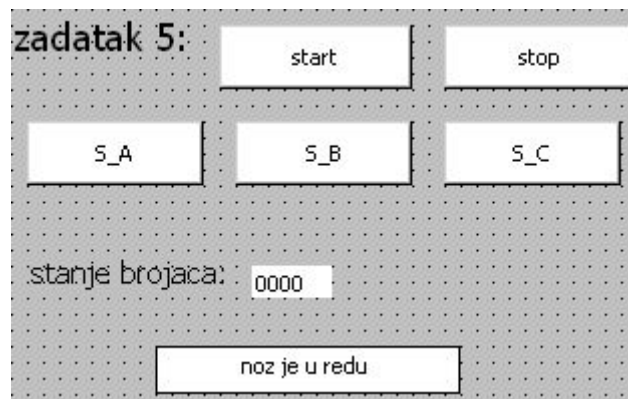
Zadatak 8

Nož se koristi za isecanje tri vrste proizvoda: A, B i C. Zbog istupljenosti, menja se posle odsecanja 1000 komada A, 500 komada B ili 100 komada C. Delovi mogu nasumično dolaziti na red za isecanje. Kada se nož istupi, aktivira se zujalica.

Za izvođenje ovog zadatka koriste se tri senzora, svaki za po jednu vrstu proizvoda, i jedan senzor koji je indikator završene operacije sečenja. Tasterom se startuje celokupan proces.

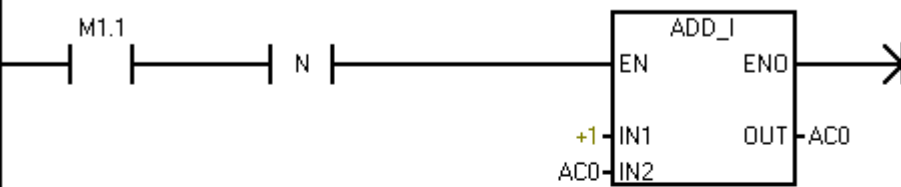
Rešenje

Na sledećoj slici je prikazan izgled aplikacije na TP-u:

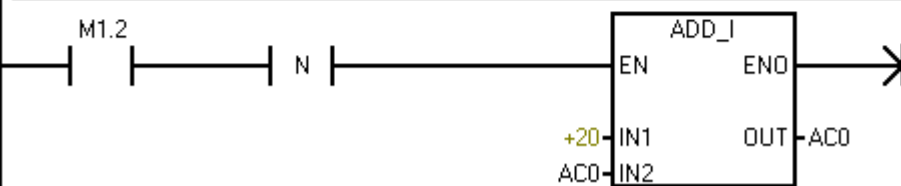


Nakon setovanja ulaza M1.0 startuje se mašina ukoliko nož nije istupljen što se može videti u (*Network 1*). Nakon registrovanja nekog od proizvoda počinje sečenje (*Network 2*). Sa gornje slike vidimo da se senzori za proizvode A, B i C setuju sa TP-a i to dugmićima S_A, S_B i S_C.

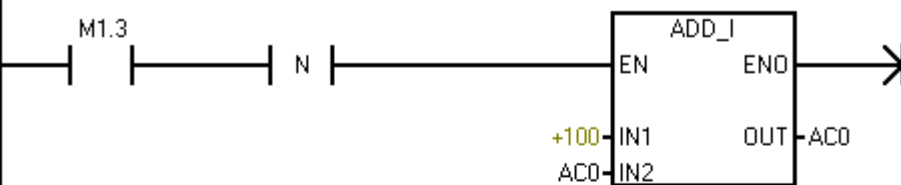
Network 3



Network 4



Network 5



Network 6



Network 7



Network 8



Slika 51.

Sada se u zavisnosti od toga koji se predmet seče dodaje unutrašnjem brojaču odgovarajuća vrednost koeficijenta sečenja: proizvod A - jedan (*Network 3*), za proizvod B - dvadeset (*Network 4*), za proizvod C - sto (*Network 5*).

Zatim se proverava da li je trenutna vrednost unutrašnjeg brojača veća od 1000, i ako jeste, uključuje se zujalica Q0.0 (*Network 6*).

Uključivanjem zujalice gasi se mašina Q0.1 radi zamene noža (*Network 1*). Nakon zamene noža pritiskamo Reset dugme I0.0 da bi unutrašnjem brojaču dodelili vrednost 0 kako bi mogao da nastavi sa odbrojavanjem (*Network 7*). Zadnji red koji je prikazan u PLC-u (*Network 8*) šalje signal na lokaciju M1.5 na koju je vezan *Symbolic IO Field* i koji nam šalje tekstualnu poruku (nož je u redu / nož treba zameniti) u zavisnosti od toga da li je lokacija setovana ili ne.

Zaključak

Iz obrađenih primera smo prvo naučili osnove rada sa TP-om i PLC-om, a kao drugo se nadamo da smo odrađenim primerima predstavili osnove rada koje će budućim kolegama omogućiti lakši početak.