

Ciklus rada PLC-a

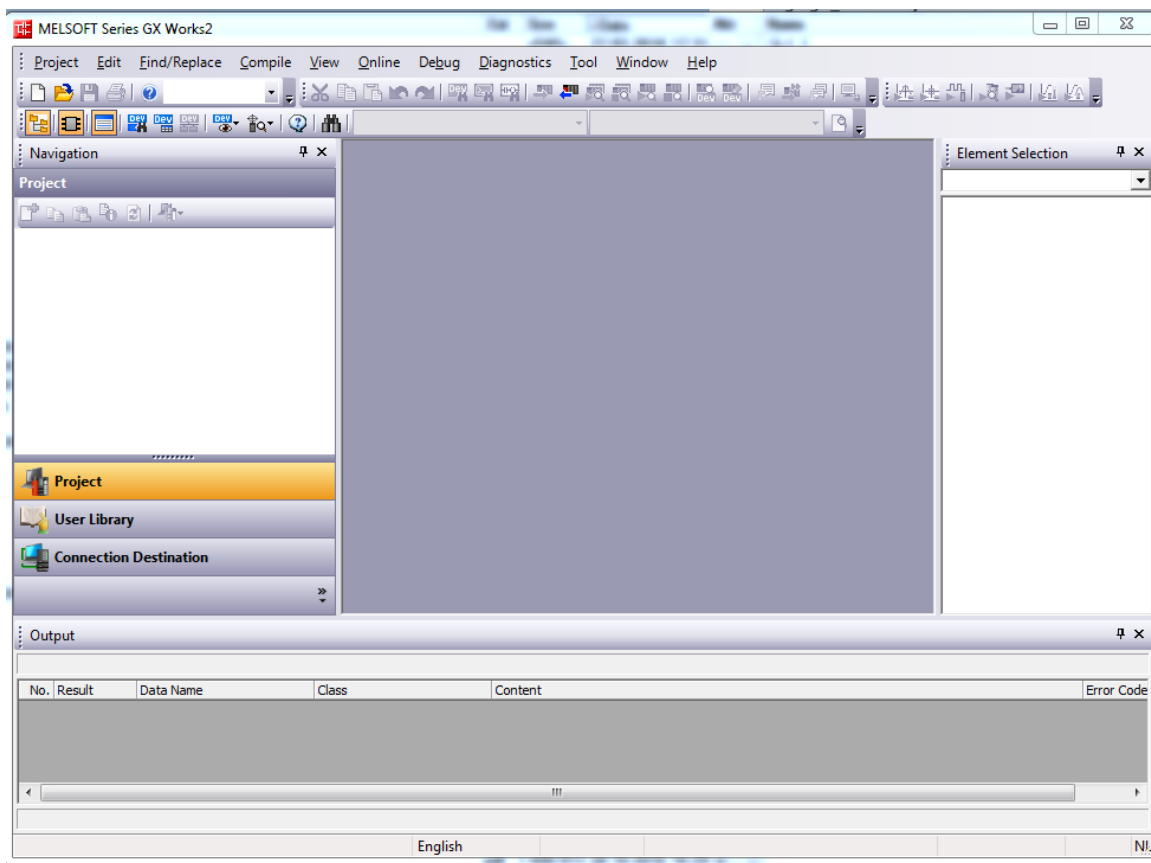
Provera da li se programski ciklus izvodi pravilno, izvodi se uz pomoć hardverskog *watchdog* tajmera, koji se proziva u svakom skenu i predstavlja osnovnu garanciju sigurnog rada. Ako se to ne dogodi signalizira se greška u samom programu ili kvar na opremi kontrolera. Na taj način štiti se sistem upravljanja, naprimer od ulaska u beskonačnu petlju. U zavisnosti od primenjenog tipa procesora u *PLC*-u, ulazni i izlazni sken izvršavaju se u vremenu reda milisekundi (0.1 do 3ms), tako da se ciklus obrade ponavlja 10 do 100 puta u sekundi. Trajanje skena ciklusa obrade, posebno programskog dela zavisi od veličine programa.

3 GX-WORKS programski paket za programiranje *PLC* kontrolera

GX-WORKS programski paket spada u aplikacije koje su namenjene za rad sa *Mitsubishi* -ovim *PLC* kontrolerima .

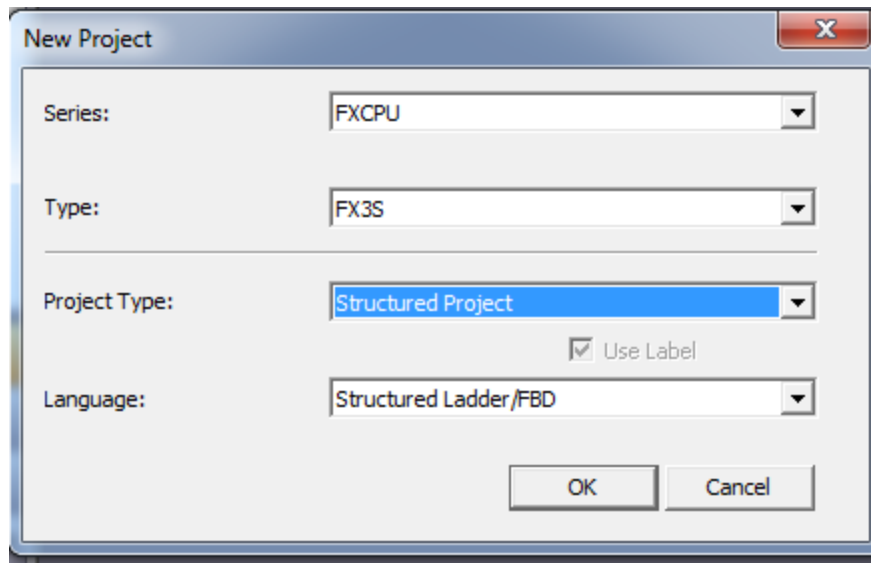
3.1 Pokretanje GX-WORKS

CX programer se poziva kao i sve druge Windows aplikacije iz "Start" menija. Po pokretanju pojavljuje se prozor kao na slici ispod.



Prvi prozor nakon pokretanja CX Programmera

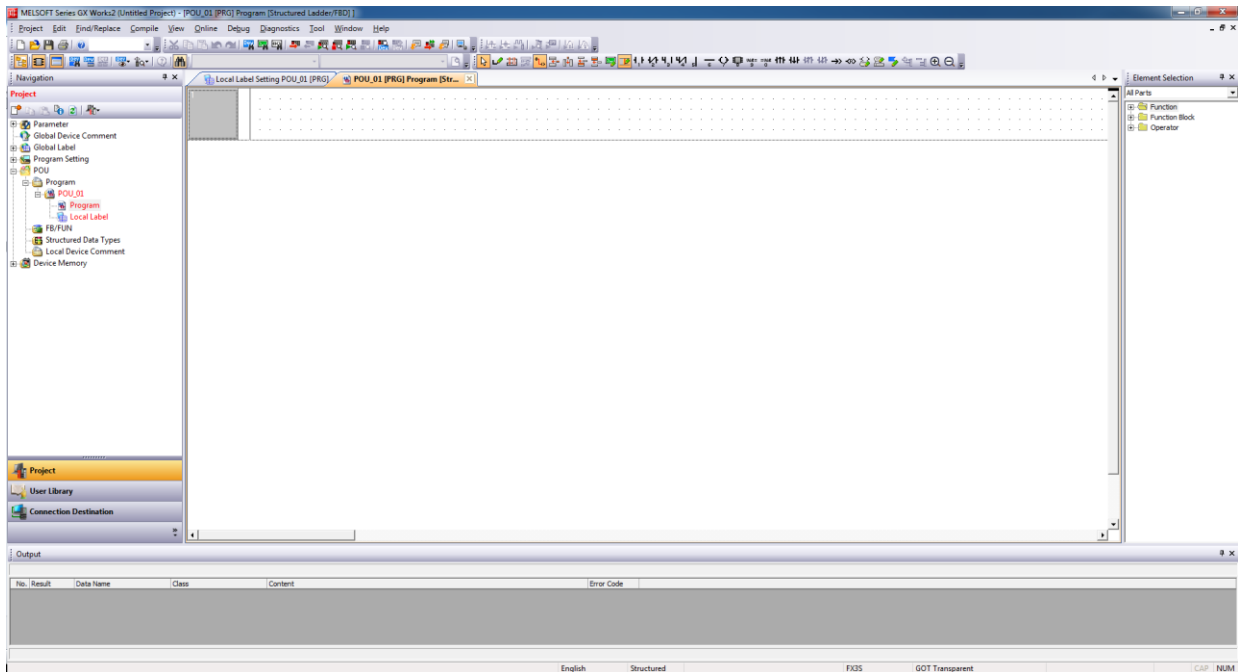
Klikom na Project - New pojavljuje se novi prozor u kome treba izabrati tip PLC kontrolera. Potrebno je izabrati FX3S tip PLC i Structured tip projekta , kao na sledećoj slici.



Podšavanje PLC kontrolera

3.2 Pisanje prvog programa

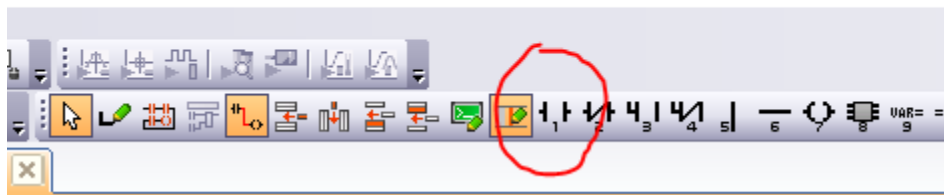
Selektovanjem *PLC*-a i klikom na *OK* pojavljuje se prozor spreman za pisanje programa kao na slici ispod.



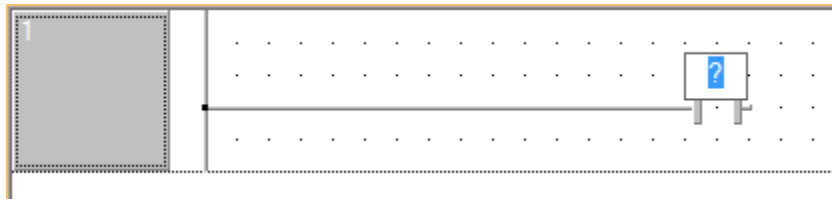
Program koji se u ovom poglavlju piše je elementaran i namenjen upoznavanju CX programera. Namena programa je da detektuje pritisak na taster i aktivira relej na izlazu *PLC* kontrolera. Dokle god je taster pritisnut i rele je aktivan, otpuštanjem tastera i rele se otvara. (rad releja i tastera se može lako pratiti preko LED dioda na kućištu *PLC* kontrolera). Ovaj jednostavan program se može napisati u svega nekoliko koraka.

1. korak

Pisanje programa počinje klikom na ikonicu broj 1 (OPEN CONTACT) u trećem redu koja simbolizuje uslov sa kojim skoro sve instrukcije počinju (prepoznaje se po dve vertikalne crte). Nakon klika kursor poprima oblik kao na crtežu sa ikonicе.

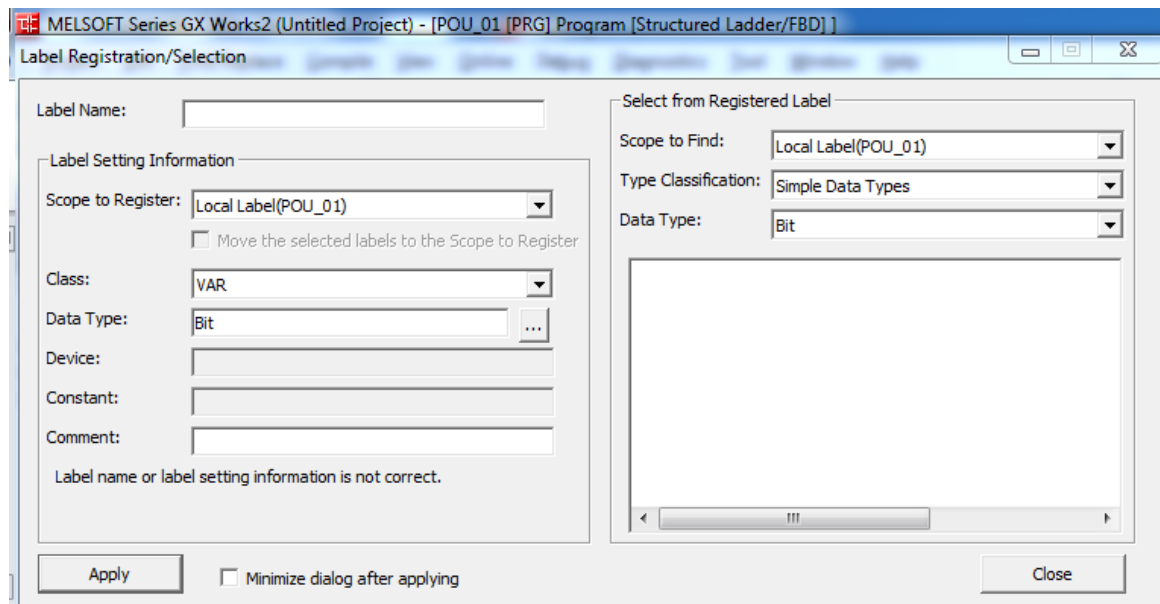


Pomeranjem na deo ekrana se leštvicom i levim klikom postavlja se kontakt na leštvicu, kao na narednoj slici.



2. korak

Ono što treba uraditi sledeće je upisivanje adrese kontakta ili promenjive kontakta. Klikom na upitnik na kontaktu pojavljuje se sledeći ekran:



U label name se upisje ime promenjive koja će predstavljati ovaj kontakt. Kod kontakta Data Type (tip promenjive) je uvek **Bit**.

Pored bitskog tipa promenjive postoje i celebrojne promenjive *WORD* i *LONG*, kao i *FLOAT* za realne brojeve. Za potrebe tajmera koristi se tip promenjive *TIME*, specifična za GX Works.

Moguće je upisati i adresu kontakta, umesto imena promenjive.

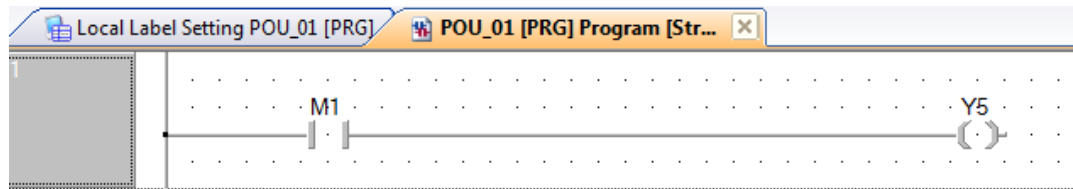
Postoje 3 značajne memorijske oblasti u ovom tipu PLC-ova:

- Interna memorija, označena sa **M**. Primer: *M1* – kontakt je zauzeo adresu 1 u internoj memoriji
- Ulazi PLC-a, označeni sa **X**. Primer: *X2* – kontakt je povezan na drugi ulaz PLC-a

- Izlazi PLC-a, označeni sa **Y**. Primer: Y0 – kontakt je povezan na nulti izlaz PLC-a

3.korak

Kontakti se još nazivaju i uslovi jer se njihovim ispunjenjem aktiviraju instrukcije koje se nalaze desno od uslovnih instrukcija. Kad je unesen uslov potrebno je uneti i odgovarajuću instrukciju koja se aktivira ispunjenjem uslova. U ovom primeru to je rele koji kontroliše izlaz Y5. Izlazne instrukcije su predstavljene sa simbolom () . Klikom na ikonicu 7 bira se opcija izlaza sa normalno otvorenim kontaktima.



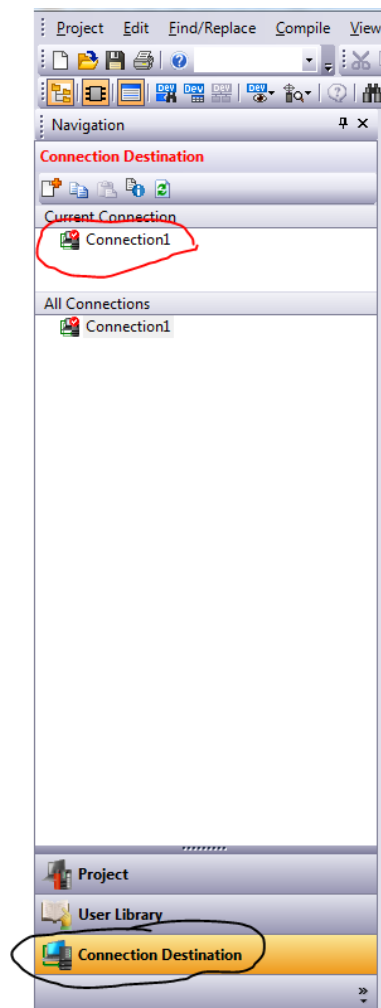
4. korak

Kompiliranje koda se vrši pomoću ikonice *Build* na slici

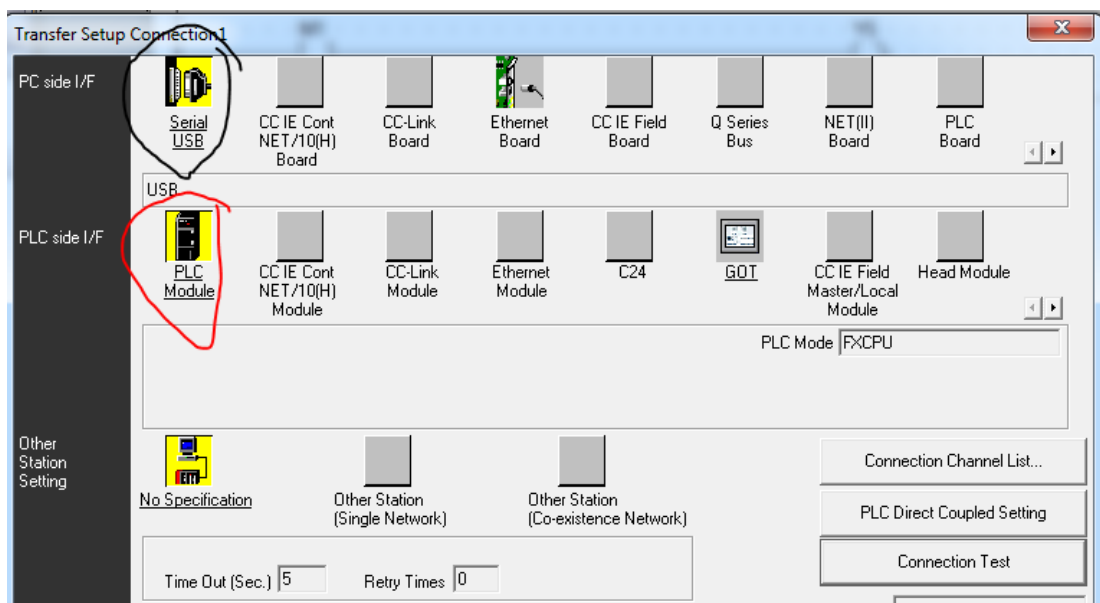


5. korak

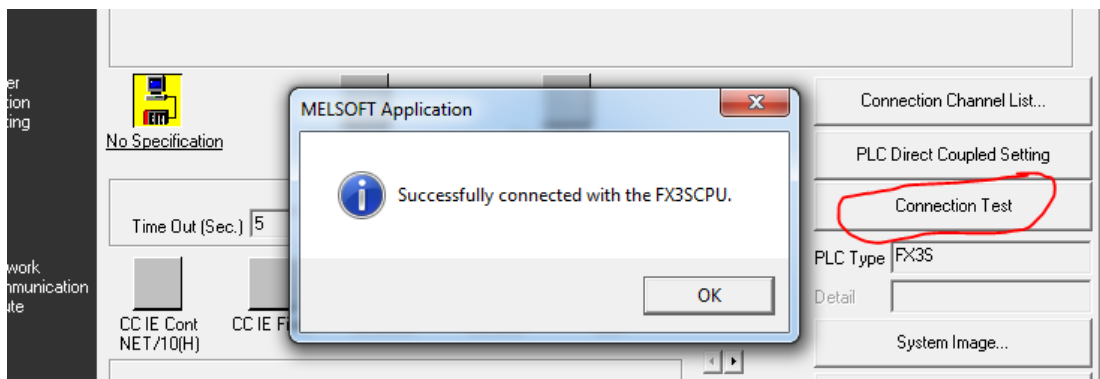
Povezivanje PLC sa softverom. Potrebno je kliknuti na *Connection destination* kao na slici, i potom kliknuti na *Connection1*.



Dobija se ekran prikazan na sledećoj slici, Potrebno je dvokliknuti na *PLC module* (zaokruženo crvenom) , tako da ta ikonica bude aktivna, tj sa žutom pozadinom. Posle toga potrebno je kliknuti na *Serial USB* (zaokruženo crnom bojom),i izabrati USB tip konekcije.



Ako je sve dobro podešeno , pritiskom na Connection test dobija se poruka od uspesnoj konekciji, kao na donjoj slici.

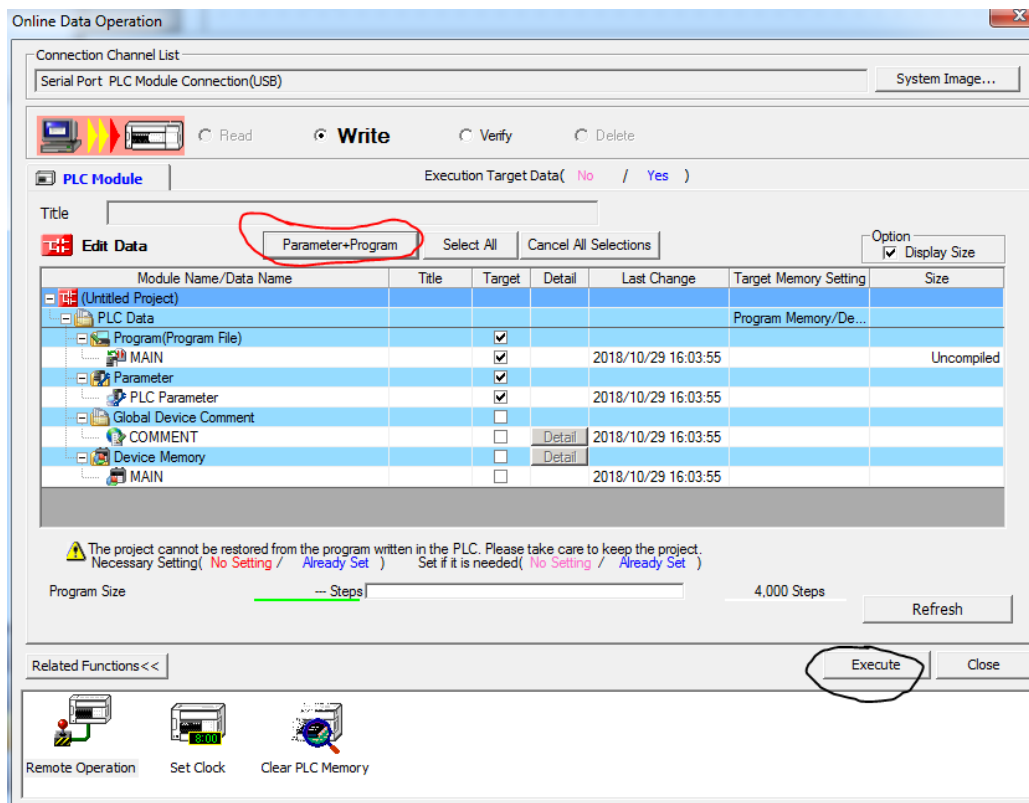


6. korak

Upisivanje koda u PLC vrši se pomoću ikonice *Write to PLC* , na slici

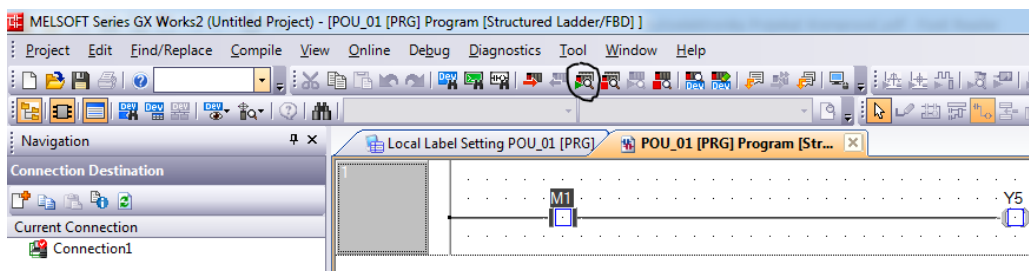


U slučaju da PLC nije povezan, ili nisu uradjena podešavanja konekcije, dobit ćete poruku da nije moguće komunicirati sa PLC-om, u tom slučaju ponovite korak 5. Ako je sve u redu, pojaviće se ekran na sledećoj slici, potrebno je kliknuti na *Parameter+program* ili *Select All*, te na *Execute* kojim se izvršava proces programiranja PLC-a.

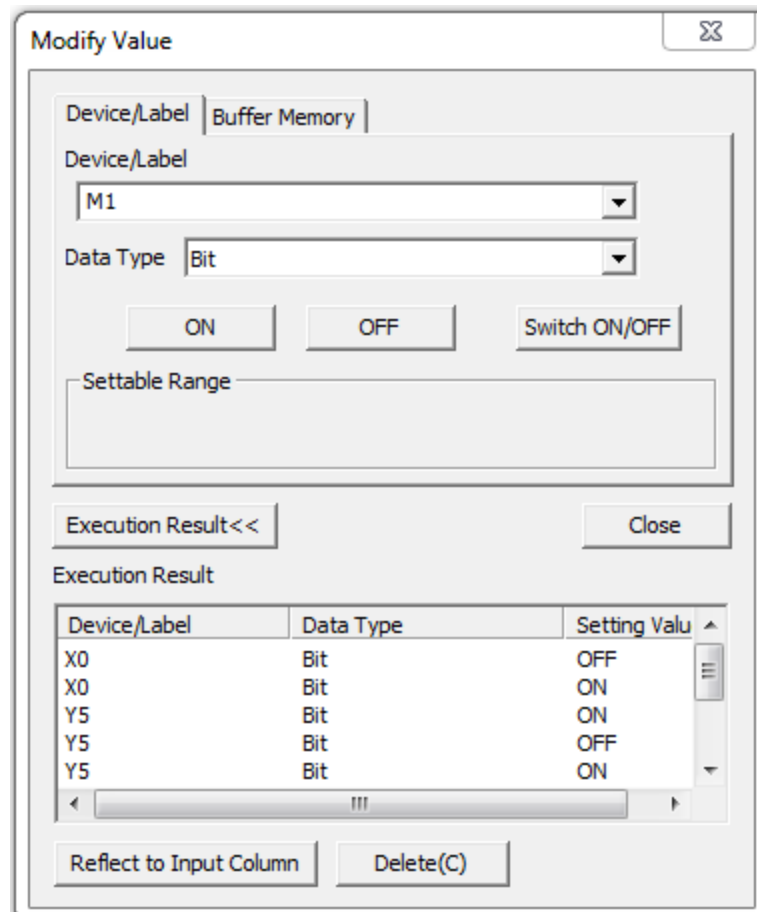


3.3 Provera rada programa

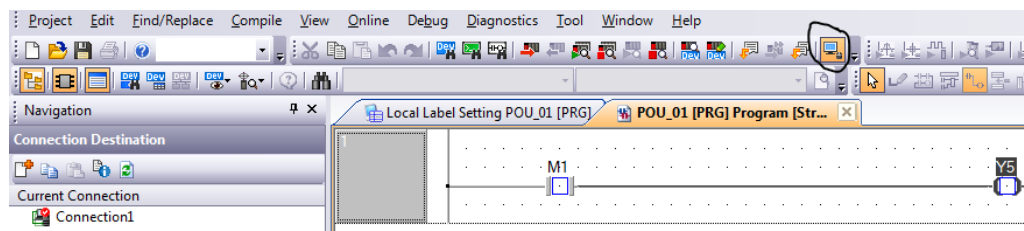
Najjednostavniji način je da se *PLC* prebaci u *RUN* mod i program isproba na samom uređaju ill u nekom sistemu automatizacije. U tom slučaju veoma je korisno *PLC* prebaiti u monitoring mod, tako da se mogu ispratiti promene na promenljivima u realnom radu *PLC*-a. Monitoring mod je indentičan *online debugging* režimu rada kod mikrontrolera. Da bi se uključio monitoring potrebno je kliknuti na ikonu *Start monitoring*, označenoj na sledećoj slici. Prekid monitoringa se izvršava sa prvom sledećom ikonicom *Stop monitoring*. Kada je *PLC* u monitornig modu, na kontaktima i izlazima se javlja plavi pravougaonik.



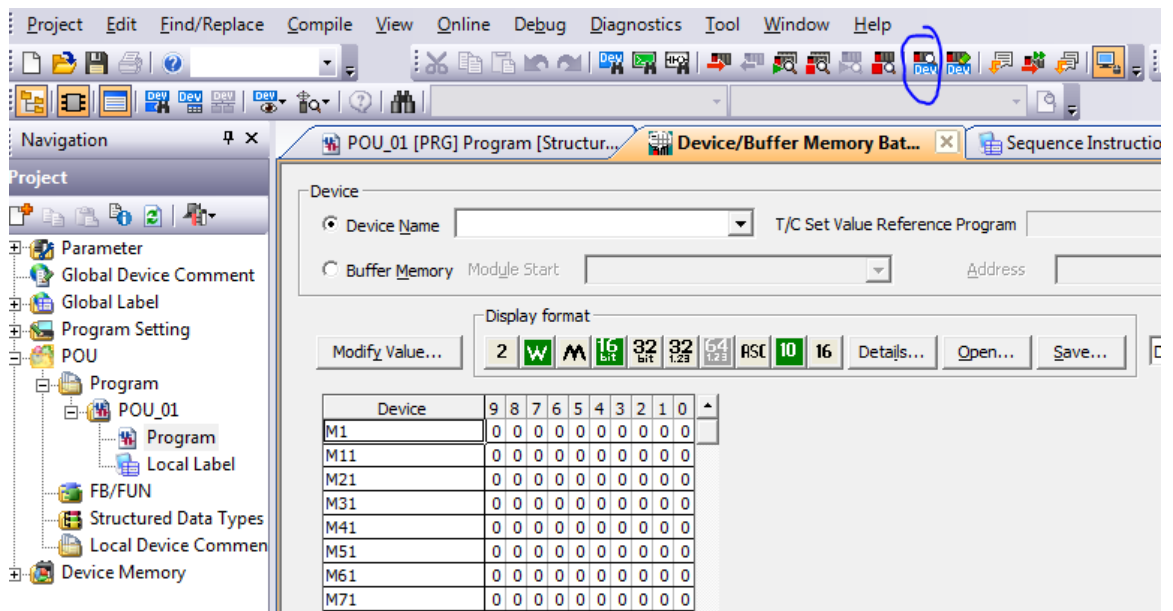
Dvoklikom na kontakt u kodu moguće je promeniti njegovo trenutno stanje, dobija se sledeći ekran gde se na pritiskom na ON i OFF ikonice forsira stanje te promenjive.



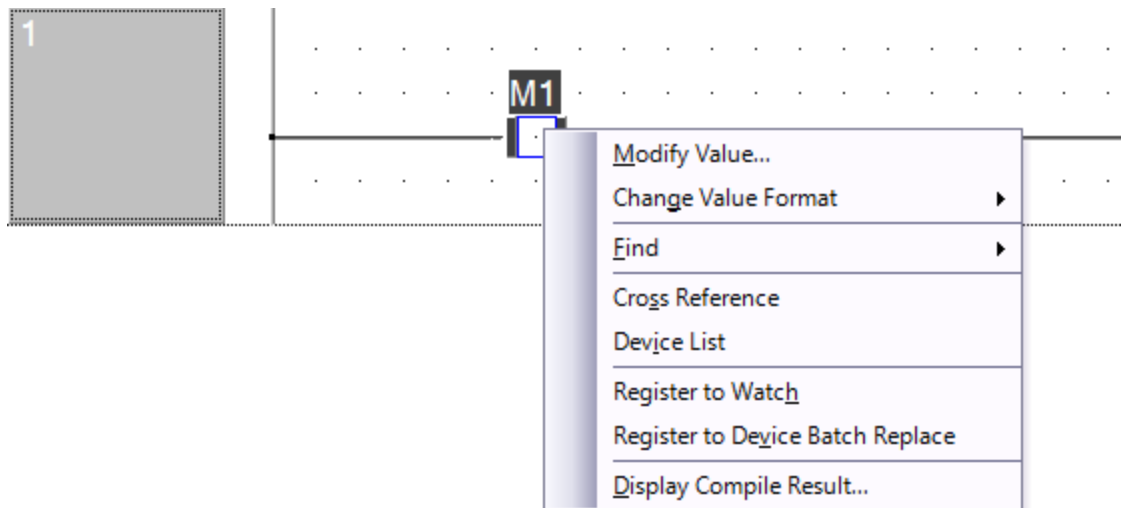
Postoji i mogućnost simulacije PLC, u kom slučaju nije potreban hardware za proveru rada softvera. Simulacija PLC se pokreće klikom na ikonicu *Start/Stop Simulation*, označenoj na sledećoj slici. Kada softver radi u modu simulacije, moguće je forsirati stanja ulaza PLC (X0, X1 itd...), što u monitoring modu nije moguće (jer tada signali na hardverskim ulazima određuju stanje).



Prozor za rad sa memorijom se dobija duplim klikom na ikonicu označenom na sledećoj slici: Otvara se novi prozor u kome je moguće podešavati koja se deo memorije prati, u kom opsegu i u kom formatu. Nije moguće pratiti promenljive po njihovom nazivu (labeli), već samo po memorijskoj lokaciji koju im je kompajler dodelio.

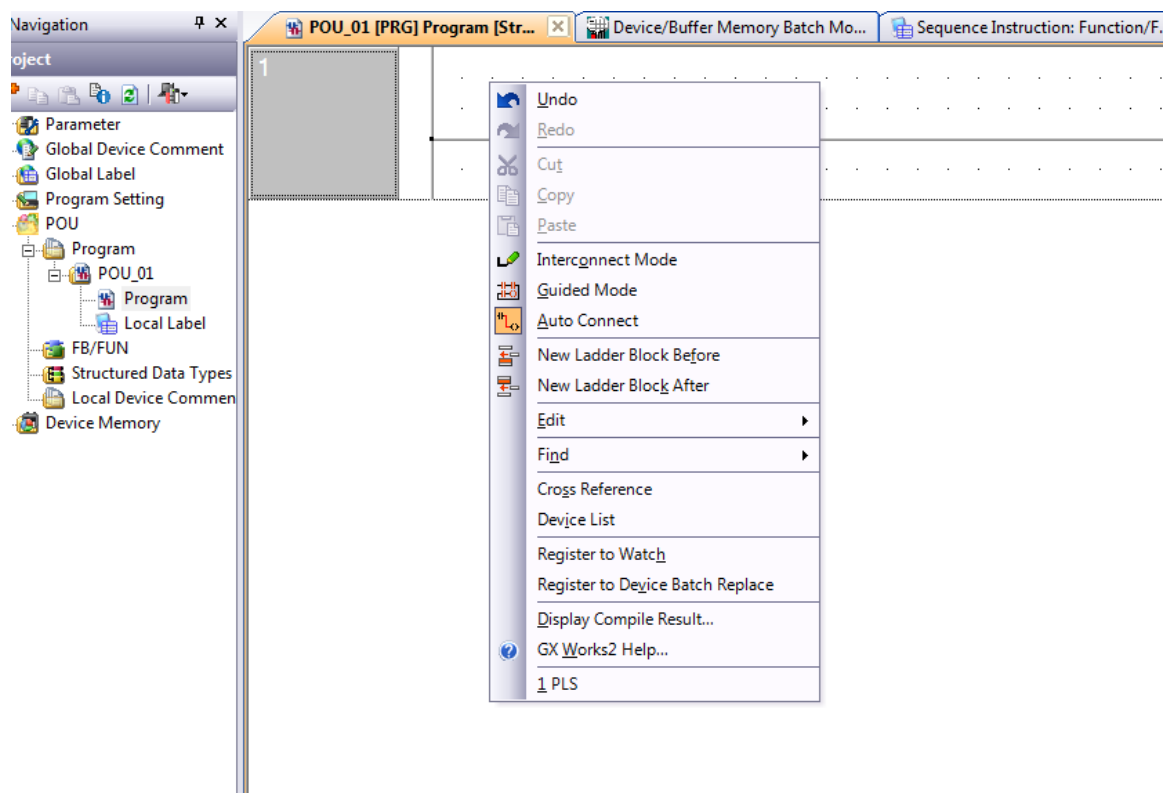


Praćenje neke promenjive po labeli, kao i po memoriskoj lokaciji, je moguće ostvariti komandom *Register to watch*, do koje se dolazi desnim klikom na promenjivu u kodu. Pri tome se otvara *Watch* prozor u donjem desnom uglu ekrana, gde se mogu pratiti promene na izabranoj promenljivoj/memorijskoj lokaciji.

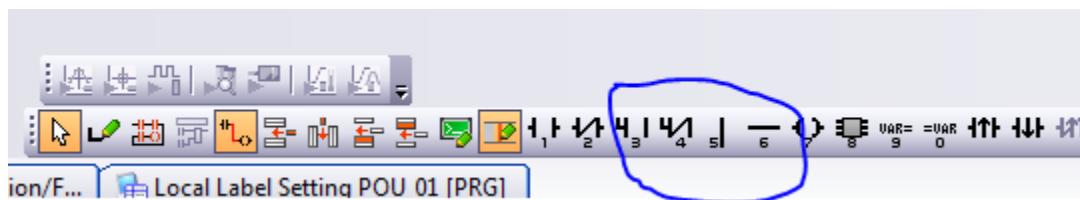


3.4 Rad sa linijama instrukcija

Osnovna funkcionalna celina jednog programa jeste *Rung* (što bi u prevodu bilo prečka ili deo lestvice) ili jednostavnije "*linija instrukcija*". Program se sastoji od više linija instrukcija koje se nalaze jedan ispod druge. Operacije sa njima se dobijaju kada se na neku postojeću liniju instrukcija klikne desnim tasterom miša i izabere opcija *New ladder block after* (ispod) ili *New ladder block before* (iznad).



Samu liniju instrukcija je nekada potrebno proširiti (kako bi se npr. stavili paralelni uslovi) što se radi sa ikonicama zaokruženima na sledećoj slici.



Samo programiranje se svodi na izbor uslova, izbor akcije ili izbor neke funkcije kao npr. tajmera ili brojača i klikom na željeno mesto u liniji instrukcija.

3.5 Primer programa - Samodržanje

Program omogućava da izlaz ostane u stanju ON i po prestanku uslova koji ga je u to stanje doveo.

Primer na slici ispod ilustruje kako se pritiskom na koji je povezan na ulaz **START** menja stanje izlaza (bita) **Y0** u ON. Otpuštanjem tastera izlaz **Y0** se ne resetuje jer preko ILI kola (koga čini sa ulazom **START**) sam izlaz **Y0** drži sebe u stanju ON u kome ostaje sve dok se ne pritisne taster na ulazu **STOPS** koji se nalazi u I vezi sa ILI kolom čime se raskida uslov i izlaz **Y0** resetuje.

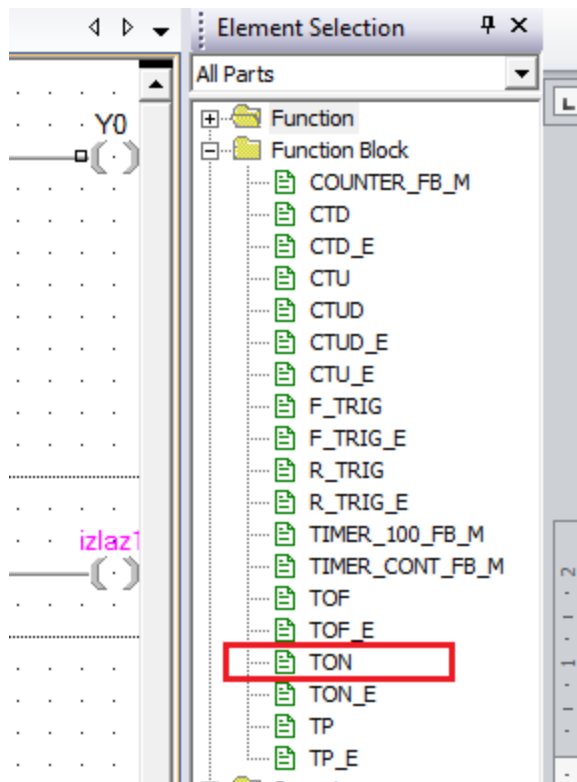
Primer samodržanja je jako čest u konkretnim primenama. Ako bi na izlaz **Y0** bio vezan neki potrošač onda bi sa dva tastera (znači bez upotreba prekidača) mogla da se ostvari START i STOP funkcija.



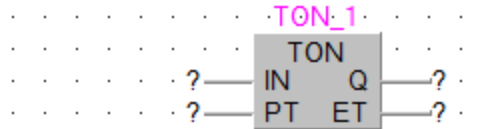
3.6 Rad sa tajmerima i brojačima

3.6.1 Tajmeri

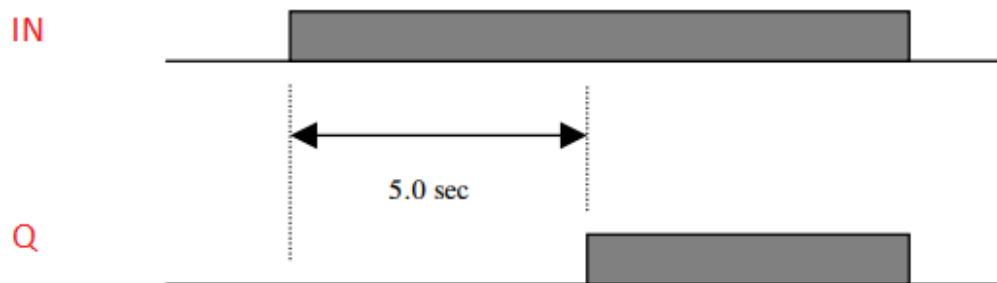
Tajmeri su složene funkcije koje imaju zadatak da vremenski odvoje dve programske akcije. U GX Works 2 tajmeri su predstavljeni kao *Function blok-ovi*. Biraju se iz prozora Element selection, najčešće se koriste tajmeri tipa TON.



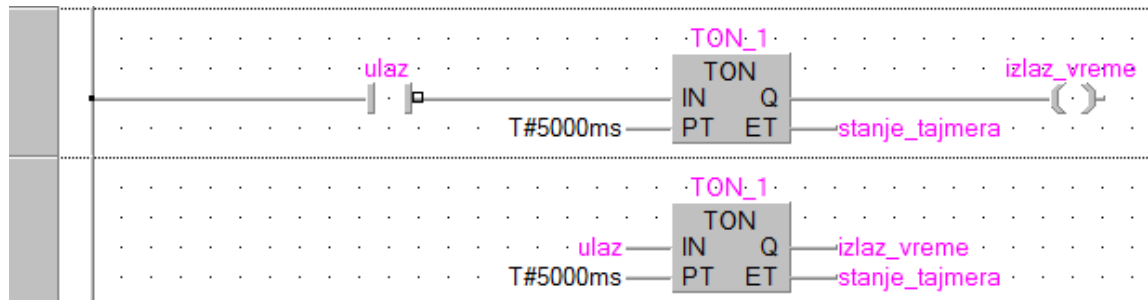
Iz ovog prozora se prevuku na glavni prozor, pri čemu se dobija instanca tajmera TON, kao na sledećoj slici.



Kada se na ulazu tajmera **IN** pojavi ON stanje, na izlazu tajmera **Q** se pojavljuje ON stanje posle vremena predefinisano na ulazu **PT**. Kada se na ulazu tajmera **IN** pojavi OFF stanje, tajmer se resetuje i izlaz **Q** prelazi u stanje **OFF**, grafički prikazano na sledećoj slici. Izlaz **ET** prikazuje trenutno odbrojano vreme tajmera.



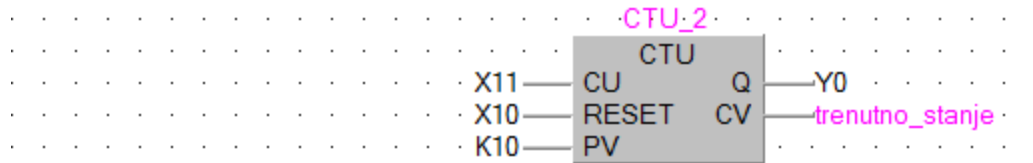
Na **IN** ulaz tajmera moguće je povezati kontakt, ili direktno upisati naziv bit promenjive umesto znaka upitnika u labelu. Isto važi i za izlaz tajmera **Q**, moguće je povezati *coil* ili upisati bit promenjivu. Obe opcije prikazane su na sledećoj slici, i obe lestvice koda predstavljaju isti kod.



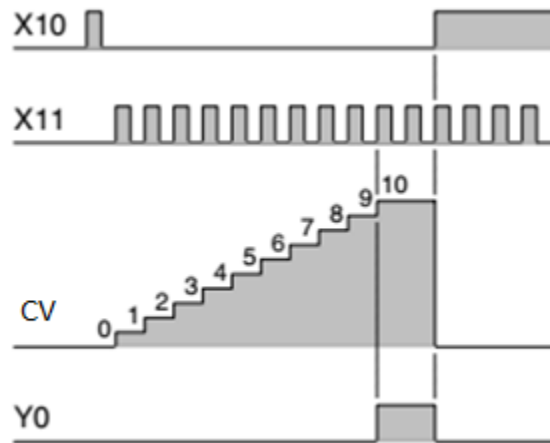
Na ulazu **PT** moguće je postaviti promenjivu tipa *TIME*, ili vremensku konstantu sa prefiksom *T#* i vreme u ms ili s, tako da *T#5000ms* i *T#5s* predstavljaju isto, pauzu od 5 sekundi.

3.6.2 Brojač (COUNTER)

Brojač je složena funkcija koja ima zadatak da broji stanja. U GX Works 2 brojači su predstavljeni kao *Function blok-ovi*. Biraju se iz prozora Element selection, najčešće se koriste tajmeri tipa CTU,CTD i CTUD. CTU predstavlja brojač tipa nagore, CTD brojač nadole, dok CTUD može da broji i nGgore i nadole, zavisno od stanja na njegovim ulazima. Kod brojača nagore prikazana na sledećoj slici, X11 je povezan na brojački ulaz **CU**, X10 na **RESET** ulaz, a na izlaz brojača **Q** povezan je Y0. Izlaz **CV** prikazuje trenutno stanje brojača.



Iz sledeće slike moguće je videti princip rad CTU brojača. Na svaku uzlaznu ivicu signala (prelazak sa OFF na ON stanje) na **CU** ulazu, brojač inkrementira svoje stanje za jedan. Kada stanje brojača dostigne vrednost podešenu na **PV** ulazu, izlaz **Q** se uključuje, tj prelazi iz OFF u ON stanje. Izlaz **Q** ostaje u ON stanju sve dok se ne resetuje brojač, tj. dok se ne dovede ON stanje na **RESET** ulaz.



4 REŠAVANJE UPRAVLJAČKIH ZADATAKA, RAD SA LADDER DIJAGRAMIMA

4.1.1 Brojanje vremenskih intervala

Ako je potrebno izbrojati koliko je vremenski ciklusa prošlo na nekom, mogu se kombinovati tajmer i brojač kao u ovom primeru.

Brojač je setovan na broji do 2000 a tajmer je postavljen na 5 sekundi što daje vremenski interval od 10.000 sekundi ili 2.77 sata. Ispunjavanjem uslova na ulazu tajmer počinje da odbrojava. Kada dođe do kraja, setuje fleg *Tiz11* koji raskine vezu i ujedno resetuje tajmer. Po isteku 5 sekundi fleg *Tiz11* menja stanje u ON i ispunjava uslov na ulazu u brojač **CTU_1**. Kada brojač izbroji 2000 takvih promena stanja flega tajmera *Tiz11* setuje svoj fleg *Counter_iz11* čime se ispuni uslov da izlaz *Y1* promeni stanje u ON. Vreme koje je proteklo između promene stanja ulaza *ukljuceno* u ON i i promene stanja izlaza *Y1* u ON iznosi 10000 sekundi.