

UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET



Vedran Mitić 18775

**SCADA SISTEM ZA NADGLEDANJE RADA
TROSTRUKOG SEMAFORA**

2025.

Sadržaj

Uvod	3
Opis projektnog zadatka	4
Glavna strana	5
Radna strana	6
1. Voz i slajder	6
2. Semafori	7
- Semafor za voz	7
- Semafor za pešake i vozila	7
3. CiCode objekti	8
4. Komandni prozor	8
Tagovi	10
Alarmi	12
Korisnici	13
CiCode	14
- Inicijalizovane promenljive	14
- Funksija dolaska voza	14
- Funkcija senzora voza	15
- Funkcija rada semafora	15
- Pomoćna funkcija za alarm	16
- Funkcija za reset voza	16
Problemi	17
Rad semafora	17
Realizacija alarma	17
Tagovi	18
Ack taster	18
Reference	19

Uvod

CitectSCADA je naš izabrani softver za monitorig i upravljanje raznim procesima i podacima koji su najčešće prikupljeni iz spoljašnjeg sveta putem uređaja poput PLC-a i dovedeni u centralni računar kroz komunikacione infrastrukture. Ovaj softver nam omogućava da se grafički i do detalja približimo samom procesu i da ga putem konfiguracija u vidu Cicoda, tagova, alarma upravljamo na željeni način. Sem obilja već ugrađenih funkcija, možemo kreirati i sopstvene. Tagovi iliti promenljive mogu biti raznolikog tipa poput analognih, digitalnih, string... Alarmi nam omogućavaju vizuelizaciju upozorenja pri određenim događajima. Takođe još jedna od prednosti SCADA-e je predstavljanje i praćenje promena tagova putem grafikona - trendovi. Pri otvaranju softvera otvaraju se tri prozora: Project Editor – za definisanje komunikacija, tagova, alarma, servera...; Graphics Builder – za dodavanje i uređivanje grafičkih elemenata na stranama; Citect Exprorer – neki vid glavnokomandujućeg prozora za orijentaciju kroz kategorije projekta.

Opis projektnog zadatka

Sistem predstavlja simulaciju automatizovanog rada tri semafora i kretanje voza na šinama. U sistem je moguće ulogovati dva korisnika: Čoveka koji radi sa samim programom(Veki, privilegija 1) i mašinovođu(Vozic, privilegija 2). Ovo je učinjeno da bi se simulirala nezavisnost u realnom svetu između te dve osobe. Postoje dve strane: Glavna strana – vid menu page-a; i Radna strana – gde se izvršava sam proces. Sistem radi na sledeći način: Najpre je potrebno ući u Glavnu stranu, ulogovati se userom Veki i kliknuti na dugme prikaz kako bi se prebacili na Radnu stranu. Na Radnoj strani u komandom prostoru prisutni tastere sa slikama semafora i voza kojima se redom omogućava rad semafora za pesake i vozila i rad semafora voza.

Videćemo da su semafori krenuli da rade na sledeći način: Najpre zeleno svetlo za pešake i crveno za vozila(10 sec), crveno za pešake(10sec) i žuto za vozila(2sec), zeleno za vozila(8sec) i tako u krug. U međuvremenu se pritiskom na dugme sa back znakom može vratiti na Glavnu stranu kako bi se promenio user na Vozic. Ponovo klikom na back dugme u gornjem levom ugлу templejta ili izborom iz padajuće liste Pages vratimo se na Radnu stranu. Sada možemo simulirati kretanje voza pomeranjem slajdera na levoj strani ekrana. Kada se voz približi putu i uđe u određeni opseg odmah se pali trepćuće crveno svetlo na semaforu voza, a ostala dva semafora takođe pale crveno svetlo. Kako su ostala dva semafora povezana i već započela malopre opisani ciklus paljenja njihovih svetala, njihov ciklus će se idalje odvijati bez obzira što gori i crveno svetlo istovremeno(objašnjeno u odeljku Problemi). Kada se ciklus završi, ukoliko je voz idalje prisutan, novi ciklus se neće započinjati već će goreti samo crvena svetla. Kada se voz udalji dovoljno, trepćuće svetlo se gasi, a ostala dva semafora nastavljaju sa normalnim radom. Kada voz stigne do kraja(vrednost 100), automatski će se vratiti u početan položaj(0) kako bi se simulacija ponovila. U sistemu postoji jedan alarm koji upozorava na neočekivano zaustavljanje voza ispred samog puta. Ukoliko se voz nalazi u određenom opsegu do 10 sekundi znači da se najverovatnije zaustavio na šinama i to ispred samog puta. Generisće se alarm kako bi ga komanda potvrdila klikom na četvrtu dugme u komadnom prozoru i javila dalje zaduženim ljudima. Ovo dugme je omogućeno i mašinovođi kako bi problem mogao da prijavi prvi koji ga primeti.

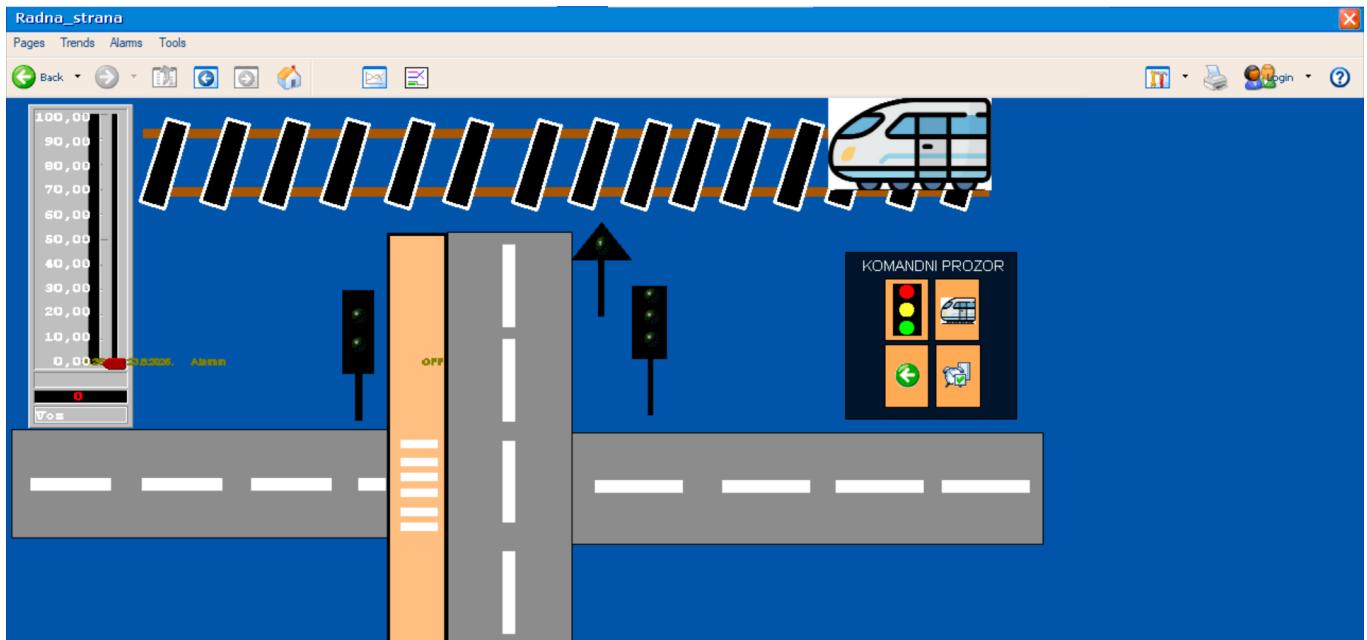
Glavna strana



Slika 1. Glavna strana

Glavna strana koristi softverski templejt kako bi se omogućilo prijavljivanje korisnika, back dugme i opadajuće liste. Na njoj je površni opis rada, slika i dugme Prikaz za odlazak na Radnu stranu. Dugme koristi softversku PageDisplay funkciju i ima privilegiju pristupa samo za nadzornika sistema(Veki).

Radna strana

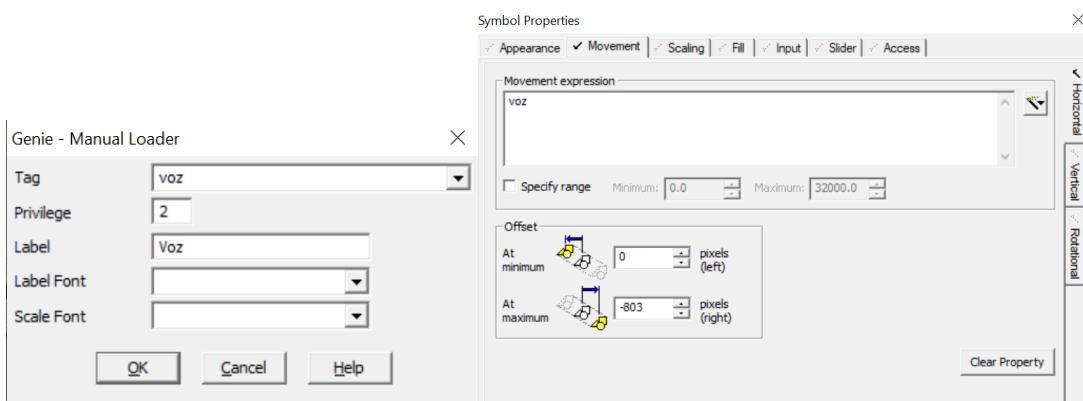


Slika 2. Radna strana

Radna strana takođe koristi isti templejt, ali iz nekog razloga komande u gornjoj traci ne rade. Na ovoj stranici se odvija ceo proces. Pored jeftinih grafičkih prikaza puta i šina imamo nekoliko bitnijih delova od kojih će svaki biti posebno izanaliziran:

1. Voz i slajder

Voz je sopstveno uvezen simbol pronađen na google images i povezan je za tag Voz. Isti tag je povezan i za slajder sa leve strane uzet iz odeljka Genie. Simbol voza je podešen tako da se kreće horizontalno u levo gde se vrednosti taga 0-100 skaliraju na 0-803 (znak minus zbog smera u levo).

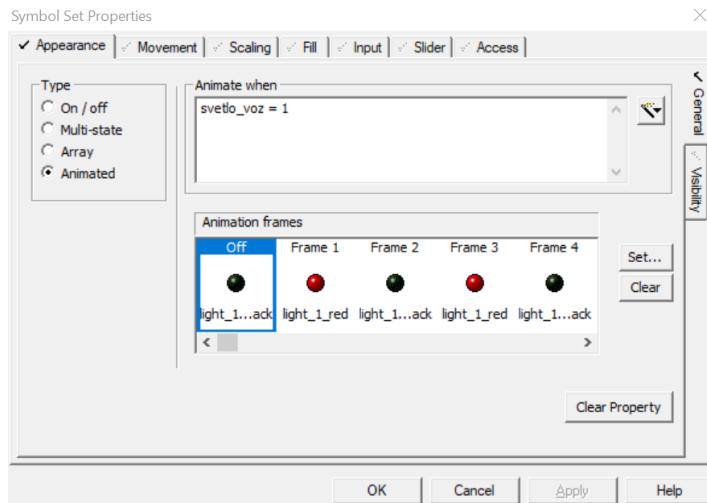


Slika 3. Konfiguracija slajdera i simbola voza

2. Semafori

- Semafor za voz

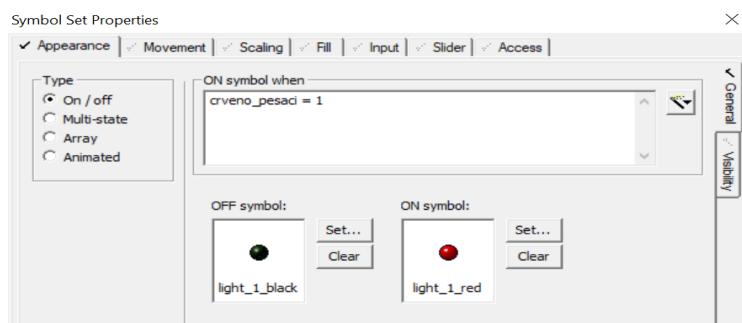
Ovde imamo jedno trepćuće crveno svetlo koje se realizuje pomoću animacije i aktivira se kada je tag svetlo_voz = 1. Ovaj tag se postavlja u CiCode funkciji `voz_dolazi()` kada voz udje u opseg (`voz > 20 AND voz < 70`), a vraća se na nulu kada voz izđe iz tog opsega što se označava promenljivom promena koja je vezana za promenljivu flag. Detaljnije objašnjenje u odeljku CiCode.



Slika 4. Konfiguracija trepćućeg svetla

- Semafor za pešake i vozila

Semafor za pešake ima crveno i zeleno svetlo koji su povezani redom za tagove crveno_pesaci i zeleno_pesaci. Kod semafora za vozila je identično uz pridodato žuto svetlo, pa su tagovi crveno_vozila, zuto_vozila i zeleno_vozila. Na slici 5. je prikazana komadana za ON stanje samo jednog od svetala zato što je konfiguracija za ostala svetla identično ista. Svetla su nameštena da se pale i gase automatizovano pomoću funkcija u CiCodu u određenim vremenskim intervalima koji su simulirani Sleep() funkcijom.



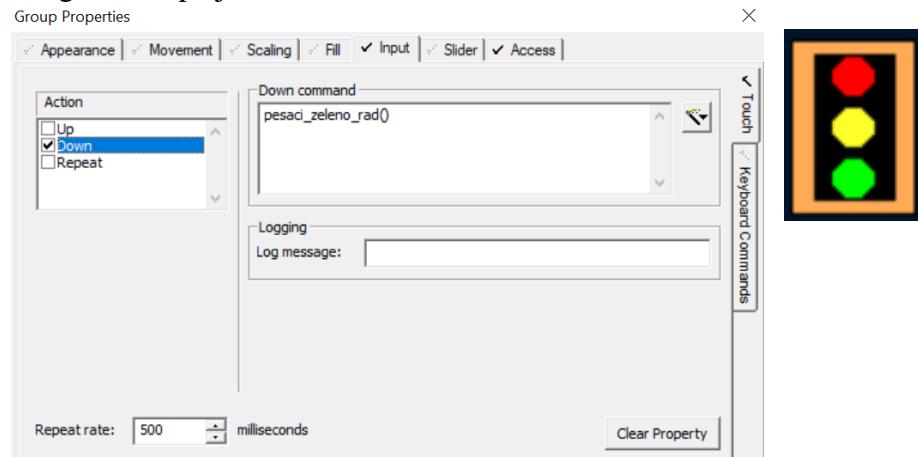
Slika 5. Konfiguracija svetala semafora

3. CiCode objekti

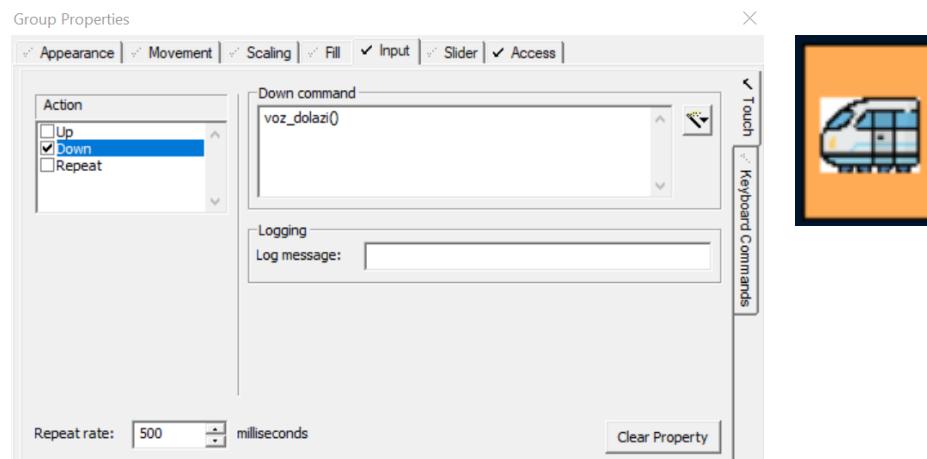
Postoje tri Cicode funkcije koje se izvršavaju beskonačno: povratak_voza() – za vraćanje voza na nultu poziciju kada dostigne maksimalnu; alarmna_funkcija() – alternativna pomoć koja je iskorišćena za realizaciju alarma(detaljnije u odeljku Problemi); flagovanje() – služi kao vid senzora lokacije voza.

4. Komandni prozor

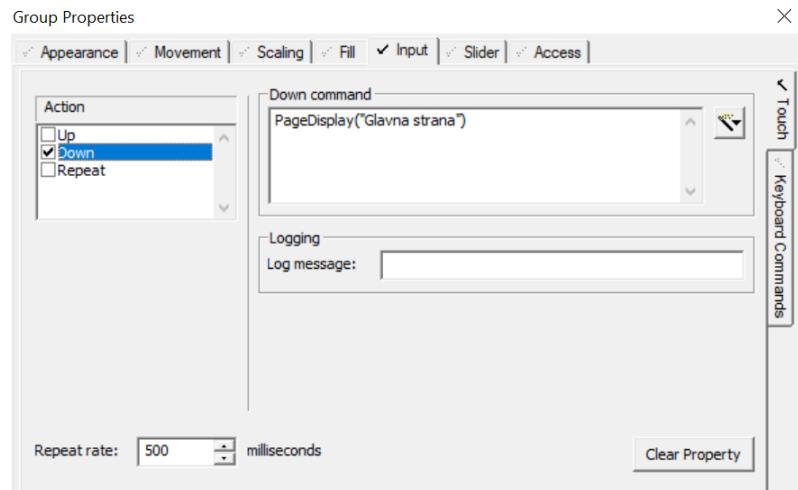
Komandni prozor sadrži četiri tastera: semafor – za start semafora za vozila i pesake, privilegija 1; voz - za start semafora za voz, privilegija 1; back – za povratak na Glavnu stranu, bez privilegije; ack – za potvrdu alarma i prosledjivanje informacije o zaustavljanju voza, bez privilegije, iz nekog razloga funkcija za potvrdu alarma ne radi pa se taster može uzeti kao imaginarna opcija.



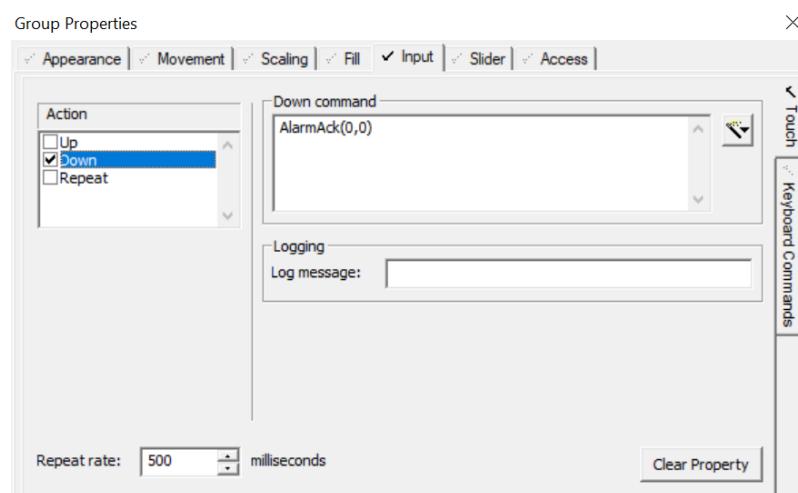
Slika 6. Konfiguracija tastera semafor



Slika 7. Konfiguracija tastera voz



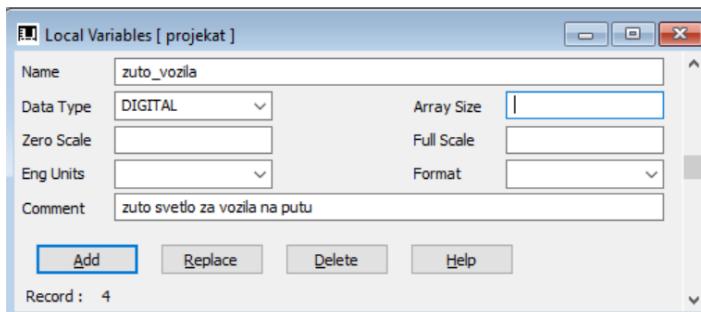
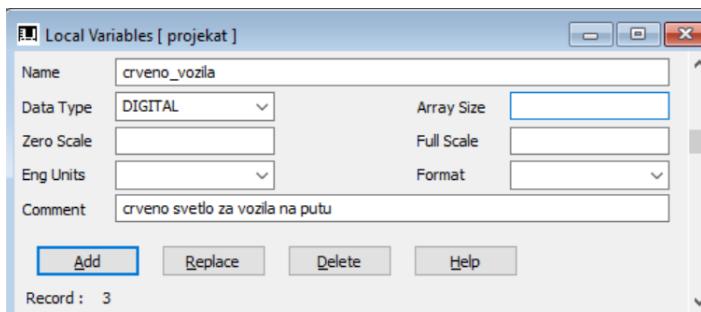
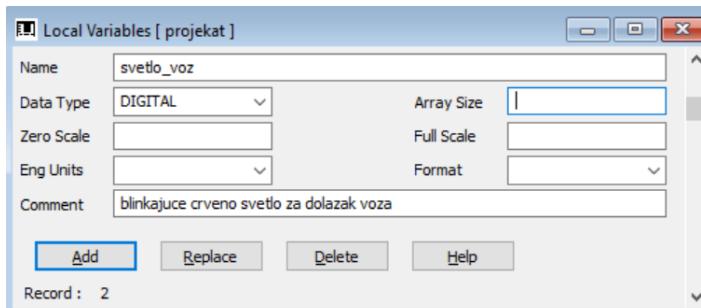
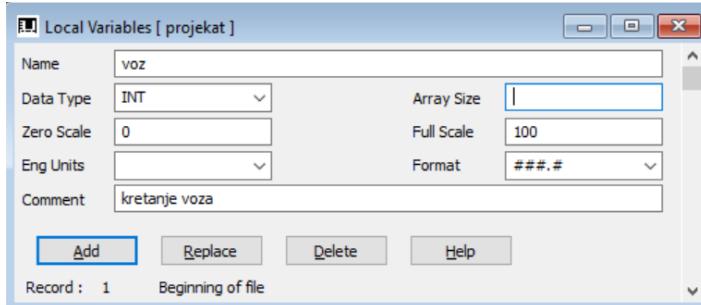
Slika 8. Konfiguracija tastera back



Slika 9. Konfiguracija tastera ack

Tagovi

U sistemu su korišćeni sve digitalni tagovi i samo jedan analogni(voz).



Local Variables [projekat]

Name	zeleno_vozila	Array Size	<input type="text"/>
Data Type	DIGITAL	Zero Scale	<input type="text"/>
Eng Units	<input type="text"/>	Full Scale	<input type="text"/>
Comment	zeleno svetlo za vozila na putu		

Add Replace Delete Help

Record : 5

Local Variables [projekat]

Name	zeleno_pesaci	Array Size	<input type="text"/>
Data Type	DIGITAL	Zero Scale	<input type="text"/>
Eng Units	<input type="text"/>	Full Scale	<input type="text"/>
Comment	zeleno svetlo za pesake		

Add Replace Delete Help

Record : 6

Local Variables [projekat]

Name	crveno_pesaci	Array Size	<input type="text"/>
Data Type	DIGITAL	Zero Scale	<input type="text"/>
Eng Units	<input type="text"/>	Full Scale	<input type="text"/>
Comment	crveno svetlo za pesake		

Add Replace Delete Help

Record : 7

Local Variables [projekat]

Name	prekoracenje_vremena	Array Size	<input type="text"/>
Data Type	DIGITAL	Zero Scale	<input type="text"/>
Eng Units	<input type="text"/>	Full Scale	<input type="text"/>
Comment			

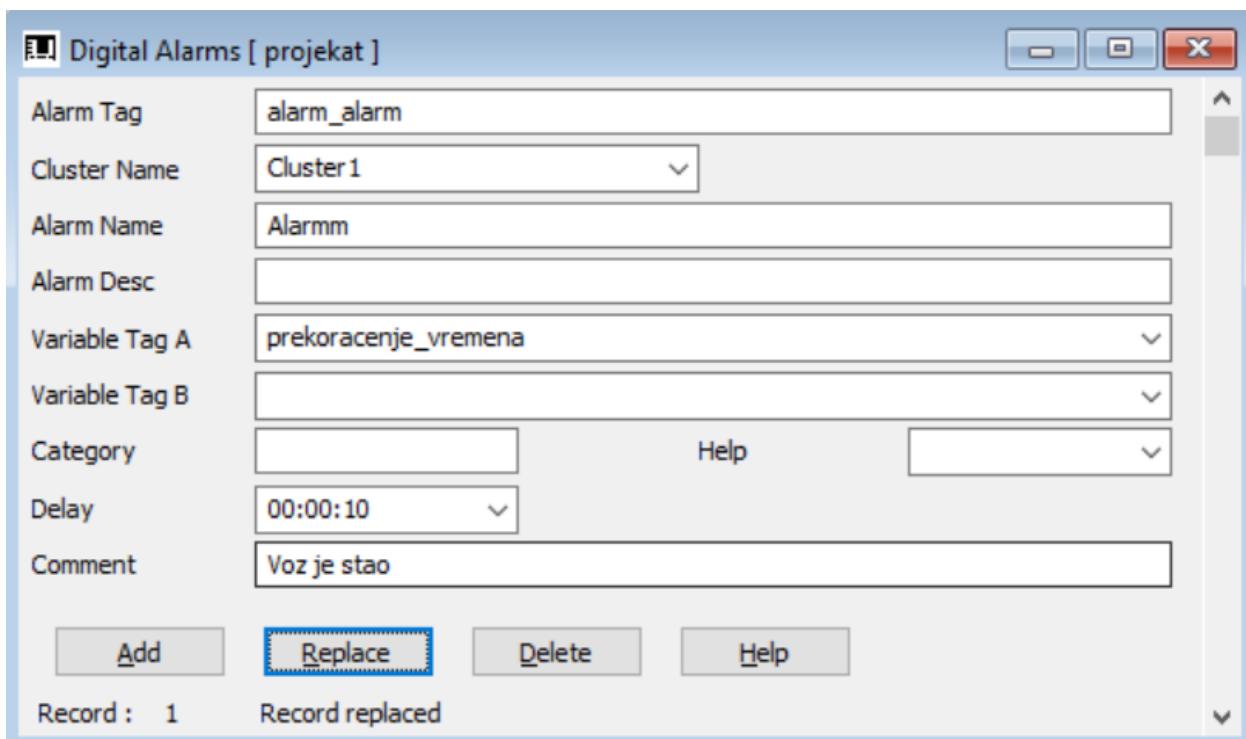
Add Replace Delete Help

Record : 8

Tag prekoracenje_vremena je digitalni tag koji je vrsta pomoći za realizaciju alarma.

Alarmi

U sistemu je korišćen jedan digitalni alarm koji je kreiran na sledeći način zbog nemogućnosti drugačije realizacije(odeljak Problemi): kreiran je digitalni alarm na slici 10. koji je povezan za tag koji smo spomenuli u prošlom odeljku prekoracenje_vremena. Ovaj tag je korišćen u Cicodu u funkciji `alarmna_funkcija()` koju smo takođe spomenuli u pododeljku Cicode objekti koja ga postavlja na 0 ili 1, odnosno pali ili gasi alarm. Sadrži delay od 10 sekundi kako bi se simulirao da je voz stao, jer on u toku tih 10 sekundi možda i izađe iz određenog opsega što znači da nije stao. Ovakvom realizacijom se malo gubi na originalnosti rada alarma, ali razlog za to će biti objašnjen u odeljku Problemi.



Slika 10. Alarm

Korisnici

Kao što je već rečeno u sistemu postoje dva korisnika Veki i Vozic.

Users [projekat]

User Name	Veki
Full Name	Vedran Mitic
Password	*****
Confirm Password	*****
Type	Upravljac
Global Privilege	1
Comment	Upravljac sistema svetala

Add Replace Delete Help Record : 1

Users [projekat]

User Name	Vozic
Full Name	Srbija Voz
Password	*****
Confirm Password	*****
Type	Simulacija
Global Privilege	2
Comment	Simulant kretanja voza

Add Replace Delete Help Record : 2

CiCode

Kod se sastoji od funkcija koje se pozivaju preko Cicode objekata i preko tastera. Funkcije koje se pozivaju tasterima se neprestano izvršavaju nakon pritisika tastera jer su realizovane u okviru beskonačne while petlje. U teoriji ovo nije potrebno zbog postojanja Cicode objekata, međutim u pokušaju takve realizacije došlo je do krešovanja programa u vidu prevelikog bagovanja, što je po mojoj proceni verovatno zbog Sleep() funkcija. Ovim je uslovljeno postojanje tastera, što sa jedne strane i nije loš izbor, jer bi recimo mogli da u nekim večernjim satima kada je saobraćaj redak isključimo rad semafora i time uštedimo na potrošnji i nepotrebnom gužvanju saobraćaja.

- Inicijalizovane promenljive

```
INT flag = 0;
INT beskonacno = 1;
INT promena = 0;
```

Promenljiva flag je ubaćena kao naznaka da li je voz prisutan ili ne, čime će se u zavisnosti od vrednosti flaga regulisati paljenje i gašenje svetla za voz kao i paljenje crvenih svetala ostala dva semafora ili njihov normalni rad. Promenljiva beskonacno služi za beskonačnu while petlju, mada i nije potrebna. Promenljiva promena služi kako bi registrovala promenu flaga i time rešila problem vraćanja svetala u prvobitni položaj pre dolaska voza.

- Funkcija dolaska voza

```
FUNCTION
voz_dolazi()

WHILE (beskonacno = 1) DO
WHILE (flag = 1) DO
IF (voz > 20 AND voz < 70) THEN

svetlo_voz = 1;
crveno_vozila = 1;
crveno_pesaci = 1;
END
END
END
END
```

Prosta funkcija koja ukoliko je voz u udređenom opsegu pali sva crvena svetla.

- Funkcija senzora voza

```
FUNCTION
flagovanje()
IF (voz > 20 AND voz < 70) THEN
flag = 1;
ELSE
flag = 0;
END
IF (promena = 1 AND flag = 0) THEN
    svetlo_voz = 0;
    crveno_vozila = 0;
    crveno_pesaci = 0;
END

promena = flag;

END
```

Ova funkcija postavlja flag u zavisnosti od toga da li se voz pribižava ili ne i vraća prethodne vrednosti svetala. Vidimo da je isti uslov za postavljanje flaga u IF funkciji i ovde i u voz_dolazi() funkciji, međutim empirijski runovanjem sistema i testiranjem pokazalo se da je flegovanje neophodno ne bi li sistem radio neočekivano u nekim situacijama. Ovakava realizacija može se posmatrati i kao duplo osiguranje od neželjenih ponašanja. Flegovanje je zamenilo inicijalnu ideju koda realizovanog pomoću case strukture.

- Funkcija rada semafora

Ova funkcija automatizuje normalan rad semafora za vozila i pešake.

```
FUNCTION
pesaci_zeleno_rad()

WHILE (beskonacno = 1 ) DO
WHILE (flag = 0) DO
|
IF (svetlo_voz = 0 AND zeleno_vozila = 0 AND zuto_vozila = 0 AND crveno_pesaci = 0) THEN
zeleno_pesaci = 1;
crveno_vozila = 1;
Sleep(10);
zeleno_pesaci = 0;
crveno_vozila = 0;
crveno_pesaci = 1;
zuto_vozila = 1;
Sleep(2);
zuto_vozila = 0;
zeleno_vozila = 1;
Sleep(8);
zeleno_vozila = 0;
crveno_pesaci = 0;

END
END
END
END
```

- Pomoćna funkcija za alarm

Kao što je već napomenuto ovo je prosto realizovana funkcija za postavljanje taga koji je povezan za alarm.

```
FUNCTION
alarmna_funkcija()
IF(voz > 30 AND voz < 50)THEN
prekoracenje_vremena = 1;
ELSE
prekoracenje_vremena = 0;
END
END
```

- Funkcija za reset voza

Vraća voz u početnu poziciju kada dođe do kraja.

```
FUNCTION
povratak_voza()
IF(voz = 100)THEN
voz = 0;
END
END
```

Problemi

U ovom odeljku navedeni su neki od problema pri realizaciji projekta, mogući uzroci i alternativni pokušaji rešenja koji takođe nisu doveli do uspešnosti. Ovaj odeljak je važan zbog praćenja same izgradnje projekta. Problemi su bili sledeći:

Rad semafora

Možda najznačajnija mana sistema jeste u Opisu projektnog zadatka spomenuta potreba za završenjem već otpočetog ciklusa semafora kada se voz pojavi. Pri pokretanju projekta to se jasno može uočiti, a najidealniji scenario bio bi da se voz pojavi kada je zeleno svetlo za vozila jer je to poslednja akcija pred završetak petlje. Ovaj problem pre svega nastaje zbog nepostojanja funkcija za rad sa vremenom(makar koliko sam uspeo da istražim). Sleep() funkcija je dobra za simuliranje proticanja vremena, ali ima tu manu što se za vreme njenog izvršavanja ne može u istoj petlji izvršiti ništa drugo. Ovaj problem sam pokušao da rešim na tri načina:

Prvi – upotrebom funkcije koja postoji kod nekih mikrokontrolerskih okruženja i koja vraća vreme proteklo od početka startovanja programa. Na taj način bi se pre svakog uključivanja svetla u neku promenljivu moglo smeštati to vreme, a onda na osnovu razlike sledećeg vremena uzetog istom funkcijom i vremena smeštenog u promenljivoj mogla ispitivati prolaznost vremena u sekundama. Na taj način bi u svakom trenutku mogao da se ispituje i položaj voza i gase svetla, dakle ne bi postojala funkcija koja to blokira. Međutim takvu funkciju nisam pronašao. Jedino što sam pronašao jeste postojanje taskova za koje nisam siguran kako i da li funkcionišu nalik taskovima u recimo RTOS sistemima.

Drugi – upotrebom brojača. Pokušao sam da ubacim promenljivu counter koja bi pre svakog stanja semafora imala vrednost nula, a onda bi u while petlji bilo ubaćen Sleep(1) i inkrement counter-a koji bi u zavisnosti od svetla za koji je povezan brojao do određenog broja(recimo za zeleno svetlo vozila od 8 sekundi do 8, ili 7 jer startuje od 0). Dakle ovo bi bio neki vid realizacije tajmera. Na ovaj način bi svetla gorela isti broj sekundi uz razliku što bi provera stanja bila moguća svake sekunde. Ovako nešto je zapravo donelo najgore rezultate gde su semafori divljali i ponašali se totalno nepovezano.

Treći – ubacivanjem IF ispitivanja stanja pre i posle svake Sleep() funkcije čime bi se problem barem minimizirao na čekanje završetka odbrojavanja Sleep-a samo jednog svetla. Ovo je delovalo jednostavno i korisno rešenje, međutim nije postiglo željeni rad, kao da je poremetilo sistem flagovanja.

Realizacija alarm-a

Ideja za alarm je od početka bila jedina opcija jer kod ovakvog sistema i nije postojao veliki broj mogućnosti, sem možda alarma za dolazak i odlazak voza za šta već postoji indikator u vidu semafora. Analogni alarm, kao prvobitna zamisao, nije mogao da zadovolji potrebu jer high i low uslovi su korisni kada se prekorače, a ovde je potrebno da se tag nađe između njih. Sledeća ideja je bila korišćenjem Advanced alarma gde je polje Expression bilo popunjeno sa voz > 30 and voz < 50, a delay bio namešten na deset sekundi. Ovakav alarm je radio na način što bi se aktivirao

odmah kada voz zakorači u taj opseg, a ne nakon deset sekundi boravka u njemu. Kao da delay nije radio kako treba. Iz ova dva razloga i nemogućnosti(neznanja) željenog ostvarenja, odlučio sam se za digitalni alarm koji je u potpunosti zadovoljio ideju na način na koji je realizovan.

Tagovi

Kako se tagovi vezuju za analogne promenljive, jedino moguće povezivanje bi bilo za promenljivu voz i time se vizuelizovalo na grafiku praćenje njegovog kretanja(pomeranja). Ovo i ne bi bilo od neke koristi za projekat, ali je ubaćeno probe radi. Međutim ispisivanje trenda na trend objectu i na trend strani uopšte nije radilo iz nepoznatih razloga. Moja je prepostavka da je problem negde u podešavanjima komunikacija(Express wizard, etc).

Ack taster

Ako se vratimo na pododeljak Komandni prozor i ack taster setićemo se da njegova funkcionalnost nema smisla. Razlog je takođe nepoznat, a za korišćenje ove funkcije vodio sam se Help-om softvera. Kako sistem i sadrži samo jedan alarm upisao sam AlarmAck(0,0), mada prepostavljam da problem može biti vezan za ovo Set Value to the AN, gde ne shvatam značenje AN.

Syntax

INT **AlarmAck**(INT Mode, INT Value [, STRING ClusterName])

Mode:

The type of acknowledgment:

0 - Acknowledge a single alarm.

- Set Value to the AN where the alarm is displayed.
- If Value is set to 0, the current cursor position will be used.

Value:

Used with Mode 1 and 2 to specify which alarms to acknowledge.

sClusterName:

Used with Mode 2 or 3 to specify the name of the cluster in which the alarms being acknowledged reside. This argument is optional if the client is connected to only one cluster containing an Alarm Server or are resolving the alarm server via the current cluster context.

This argument is not required where:

- the mode is 2 and the value is 255 (hardware alarm category).

This argument is enclosed in quotation marks "".

Pojedine probleme bezuspešno sam pokušao da rešim i importovanjem projekta u noviju verziju 7.5.

Reference

Ideja za sam projekat je originalna i prethodno razvijena za projekat za PLC u kome su neki od prethodno navedenih problema lakše prevaziđeni. Literatura koja je korišćena su predmetne prezentacije, Help softvera i YouTube tutorijali:

https://www.youtube.com/watch?v=4petUp7WNAc&list=PLlprgxTm3rViawDcg3o5AUmHaRI_EhZIAL

https://www.youtube.com/watch?v=IWu-4g3EE_s