Internet Stvari

Projektna dokumentacija za 1. Kolokvijum

1.Uvod

Cilj projekta je realizacija sistema za upravljanje semafora uz pomoć 2 Arduino Uno (Rev3) mikrokontrolera, 7 dioda i 2 tastera. Jedan arduino upravlja radom semafora za vozila i pešake, dok je drugi nadsistem koji može aktivirati ili deaktivirati sistem semafora.

S obzirom da je promena stanja na semaforu tako da bude zeleno za pešake a crveno za vozila realizovano pritiskom pešaka na taster, u programu je definisana **attachInterrupt()** funkcija unutar **setup()** funkcije koja se odnosu na prekidnu rutinu pritiskom na taster i ovako je definisano za oba arduina.

S tim što će kod prvog arduina zbog čekanja da prođe 10 sekundi za vozila, tokom prekida stiska biti setovana logička promenljiva koja će se proveriti nakon isteka 10 sekundi, dok će na drugom arduina pritisak na dugme odmah invertovati stanje sistema.

Serijska komunikacija je setovana na serijski port sa ***Serial.begin(9600)*** kao najnižim nivoom frekvencije (koliko ćemo bitova slati po sekundi) u serijskoj komunikaciji s obzirom da ne šaljemo veliku količinu podataka.

Na prvom arduino se u posebnoj funkciji za proveru zahteva proveravala ima li dolazne poruke na serijskom portu za prijem komandom ***Serial.available() > 0****.* Zatim pomoću funkcije **Serial.readString()** učitavamo string sa izlaza drugog arduina.

Drugi arduino šalje na izlaz poruke pomoću naredbe ***Serial.println(stanje);***.

Umesto tajmera je korišćena funkcija ***delay(interval);***kojom se Arduino sistem pauzira na određeno vreme. Zato imamo proveru na prvom arduinu nakon svakog ***delay()*** da li ima promene stanja sistema, kako bi se što pre zaista promenilo stanje na prvom arduinu.

Sistem ima 2 glavna stanja (aktivan, neaktivan), i unutar aktivnog stanja ima još 5 podstanja.

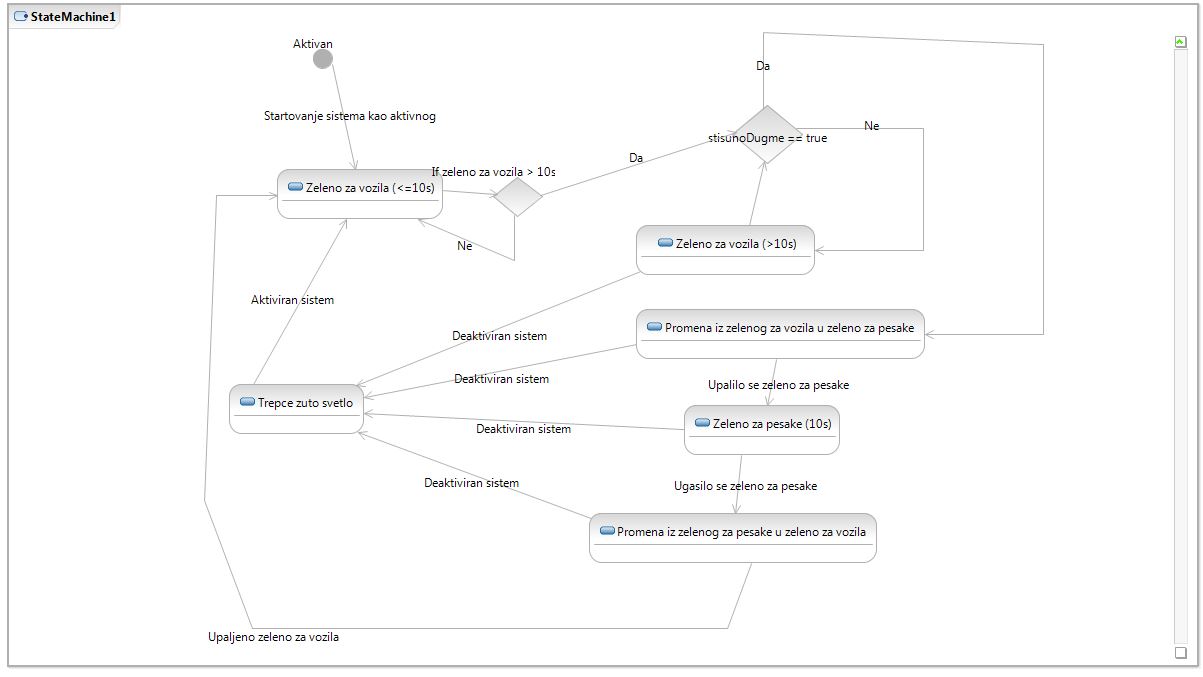
Dijagramom stanja je prikazan prelaz iz jednog u drugo stanje pod određenim uslovima.

2. Dijagram stanja

Ceo sistem funckioniše u 2 glavna stanja kojima upravlja drugi arduino koji šalje poruke i koji je transmitter, a drugi prijemnik. Na prvom arduinu imamo za aktivno stanje i još 5 podstanja ( 2 podstanja dok je zeleno za vozila (pre i posle 10s) i 3 podstanja za smenu svetla (promena iz zelenog za vozila u zeleno za pešake, zeleno za pešake i iz zelenog za pešake u zeleno za vozila).

Na početku je ssistem aktivan i ulazi u stanje prvih 10 sekundi upaljenog svetla za vozila. Nakon 10 sekundi se proverava da li je dugme stisnuto, ako jeste onda ide sekvenca paljenja i gašenja svetla za pešački prelaz. Ako nije stisnuto, nastavlja se sekvenca zelenog svetlja sve dok se ne stisne dugme za pešački ili deaktivira sistem. Sistem se može deaktivirati i u jednom od 3 podstanja dok se realizuje procedura smene svetala.

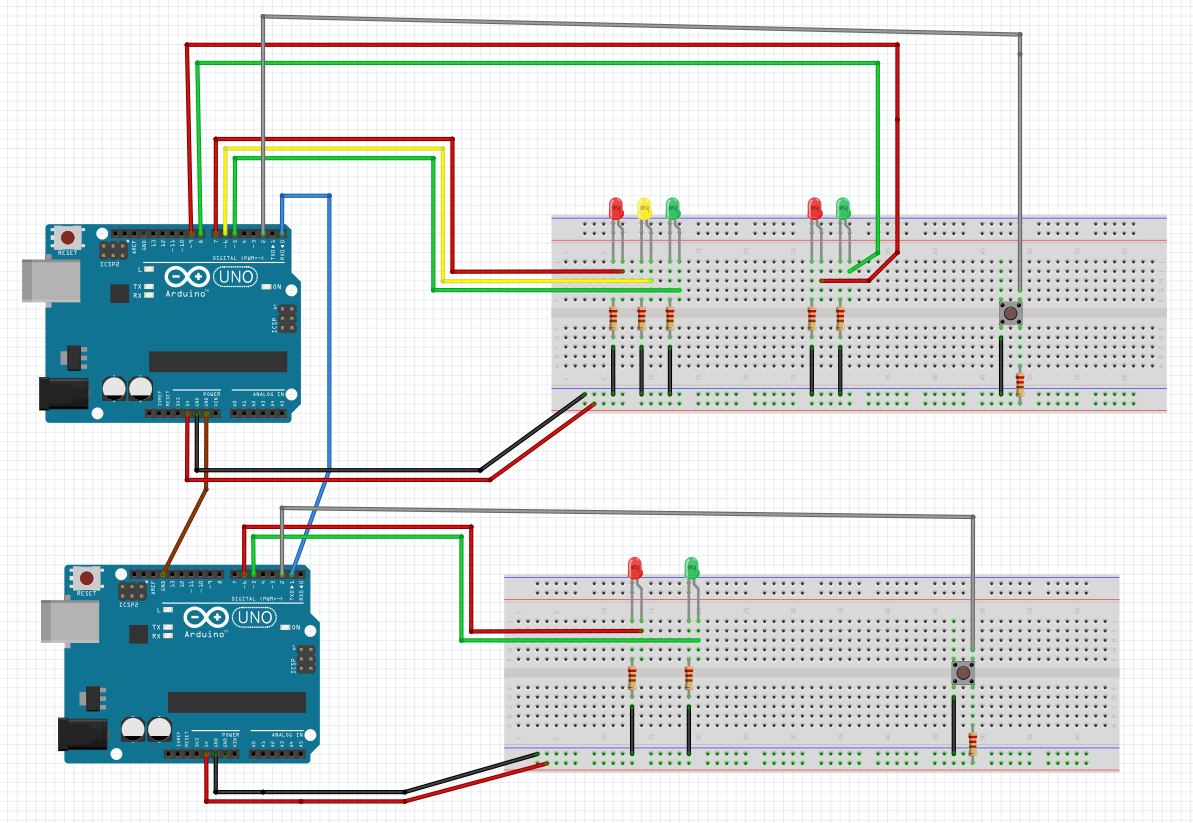
Kada se deaktivira sistem, onda samo trepce žuto svetlo, i pri ponovnoj aktivaciji je sistem u stanju prvih 10 sekundi zelenog svetla za vozila.



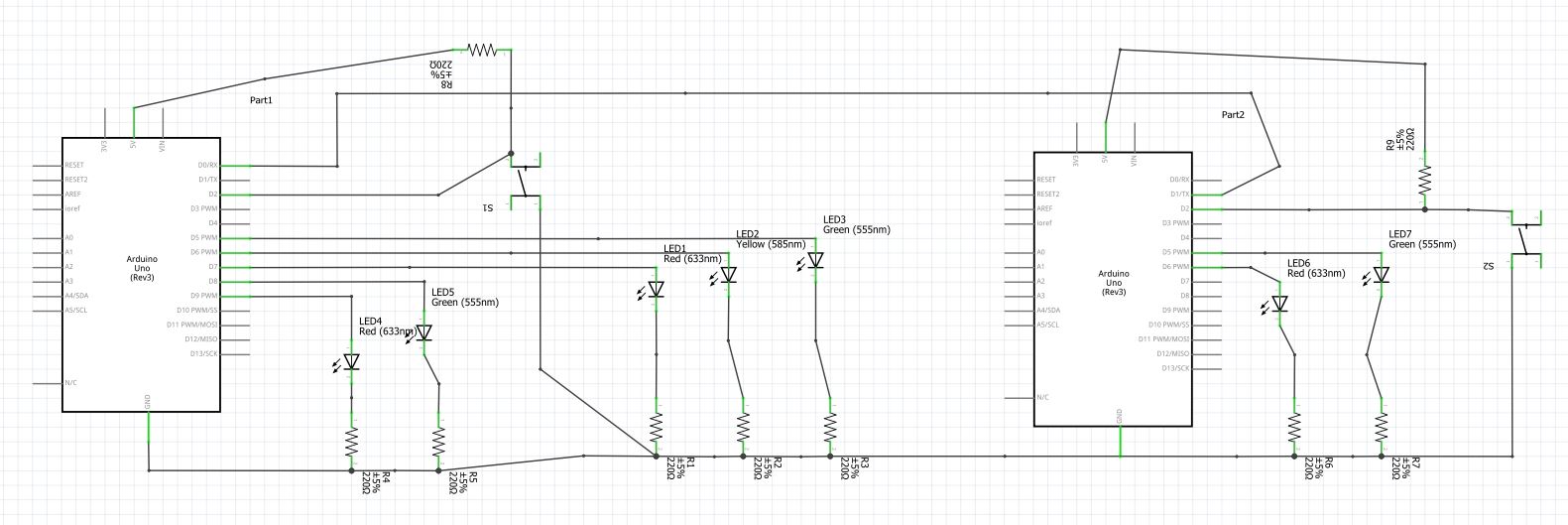
3.Fizička realizacija i šema električnog kola

2 Arduina su povezana jednom žicom za prenos podataka, jer je komunikacija tipa Simplex (Ardunio za (de)aktivaciju sistema je transmitter, dok je Arduino za upravljanje semaforima isključivo prijemnik). Takođe smo povezali oba ardunia preko GROUND pinova da bi izjednačili potencijal na oba mikrokontrolera. Za sve diode i dugmiće smo koristili otpornike od 220 oma.

Dugmići su povezani na pinove D2 kako bi radili na običan klik bez zadržavanja stiska.



Na katode dioda smo doveli otpor od 220 oma za napon. Dugme koristi isti otpornik za napon.



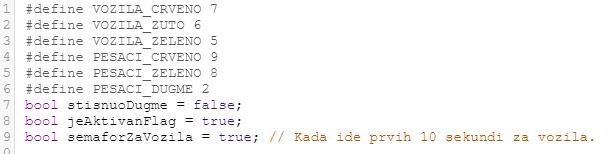
4. Analiza koda programa

Ispod je prikaz delova koda prvog pa zatim drugog arduina od vrha ka dnu fajla.

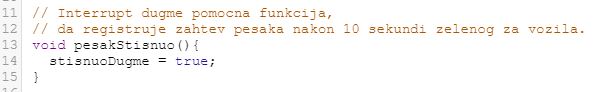
**1.Arduino**

Definisanje globalnih konstanti za ulaze diode za vozila, pešake i dugmeta na određene digitalne pinove, kao i pomoćnih promenljivih za upravljanje promenom stanja.

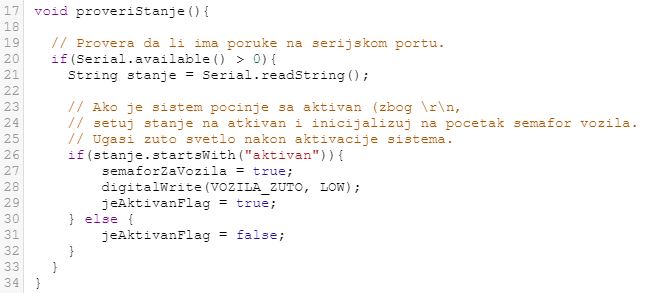
Promenljiva ***stisnuoDugme*** registruje stisak pešaka i u interrupt se setuje na true, pa se odloži ako je potrebno sačekati da prođe 10s za vozila. Zatim imamo promenljivu **jeAktivanFlag** koja služi za promenu između 2 glavna stanja sistema (aktivan, neaktivan) i treća promenljiva **semaforZaVozila** je true ako tek treba sačekati 10 sekundi za vozila, kako nakon 10 sekundi ako pešak ne stisne dugme ne bi opet morao da čeka 10 sekundi.



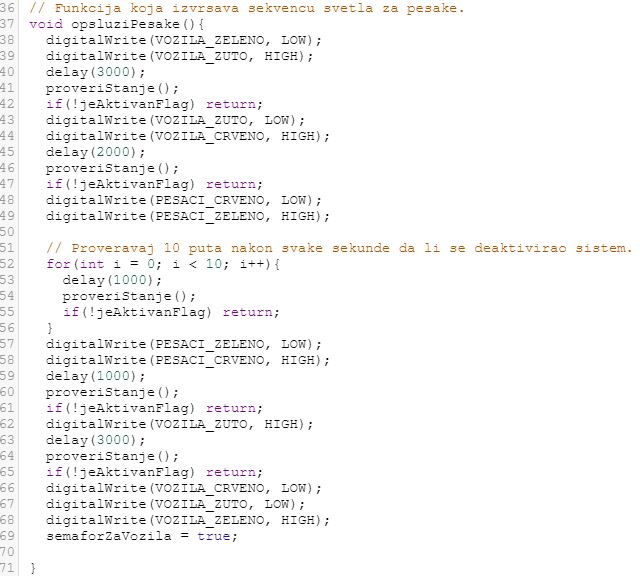
Funkcija koja se poziva nakon interrupt i setuje globalnu promenljivu stisnuoDugme na true kako bi kasnije nakon 10 sekundi proverili da li treba pustiti zeleno za pešake.



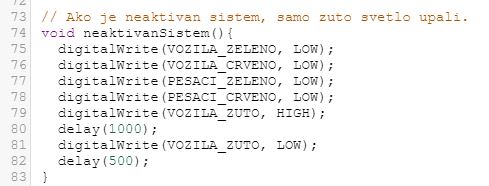
Funkcija koja proverava da li ima promene stanja na serijskom portu (da li ima nove poruke), i samo ako ima onda učita novo stanje pomoću ***Serial.readString()*** i radi proveru sa ***startsWith()*** funkcijom kako bi ignorisao praznine za novi red. Ako je sistem aktivan onda pored toga što menja ***jeAktivanFlag***, ako je aktiviran sistem onda isključi žuto svetlo koje je treptalo za neaktivan sistem i setuj ***semaforZaVozila*** na true kako bi krenulo ponovno odbrojavanje 10s zelenog za vozila.



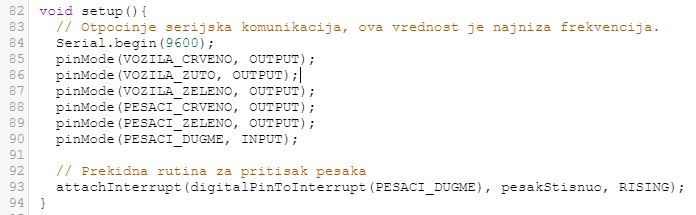
Ispod je prikaz funkcije koja izvršava sekvencu paljenja i gašenja određenih diode uz određen delay i provere da li ima promene stanja sistema nakon svakog delay, kao i izlazak iz funkcije ukoliko je sistem deaktiviran.



Kada je nekaktivan sistem onda pogasi sva svetla i podesi treptanje žutog svetla tako da je 1s upaljeno, a 0.5s je ugašeno.

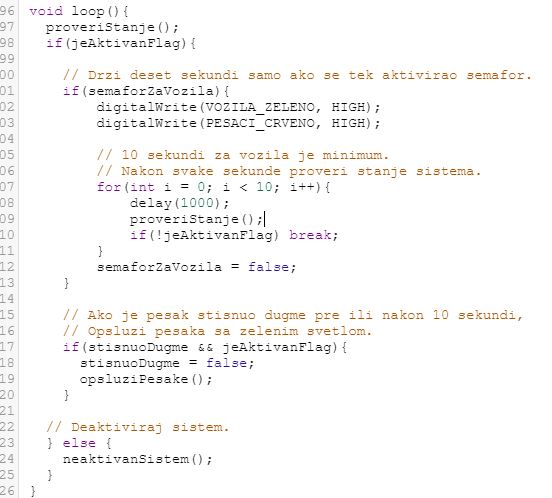


Pri paljenju sistema smo definisali tok serijske komunikacije sa 9600 bitova u sekundi frekvencijom kao pinMode za diode na izlazni mod, dok je dugme samo ulazni mod za struju, kao i sistemski prekid za pritisak dugmeta i poziv fukcije **pesakStistnuo**.



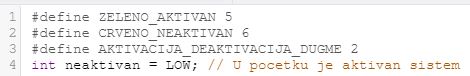
U glavnoj petlji uvek na početku proveravamo ima li promene stanja sistema. Na početku sistema je setovano tako da je aktivan sistem. Ako je neaktivan sistem samo će se izvršiti funkcija u else bloku na dnu. Ako je aktivan sistem onda proveri da li treba izvršiti početnu seriju od 10s zelenog svetla za vozila. Ako treba onda nakon svake sekunde proveri ima li promene stanja sistema i izađi iz petlje ako je deaktiviran sistem.

U drugom if bloku se proverava da li je pešak stisnuo dugme i da li je aktivan sistem, pa ako jeste onda izvrši seriju promene svetla i zelenog svetla za pešački prelaz.

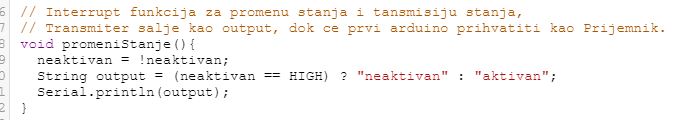


**2.Arduino**

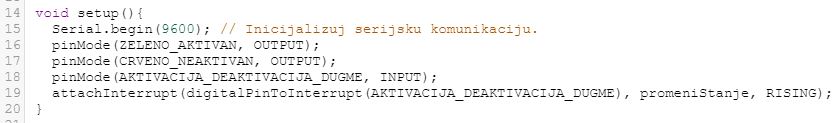
Definisali smo globalne konstante za diode i dugme i pomoćnu promeljivu neaktivan koja će služiti i za promenu stanja sistema i za paljenje diode.



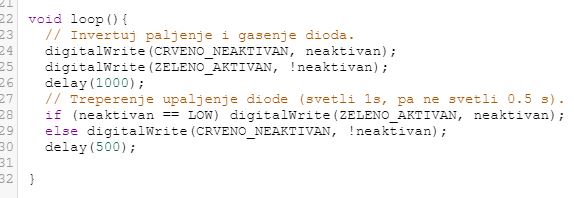
Funckija koja se okida na prekid stiskanjem dugmeta i invertuje stanje sistema i u zavisnosti od toga da li je LOW ili HIGH vrednost, šalje poruku što će učitati prvi arduino kao promenu stanja.



Na početku je inicijalziovana serijska komunikacija sa 9600 bitova po sekundi, pinovi sa izlaznim modom i dugme sa ulaznim i nakačen je sistemski prekid za dugme koji poziva funkciju **promeniStanje**. Prvi parameter prekida je broj prekida i uglavnom se koristi digitalPinToInterrupt da za određeni broj pina prebaci u odgovarajući broj sistemskog prekida, dok je poslednji parametar je mod kada se dešava prekid i u ovom slučaju je to kada se stisne i pusti dugme (iz LOW u HIGH stanje).



U glavnoj petlji u zavisnosti od vrednsoti neaktivan koja se invertuje nakon prekida će uvek jedna dioda da treperi (svetli 1s i biti ugašena 0.5s), dok će druga dioda biti ugašena.



5. Zaključak i moguća nadogradnja

Ovaj sistem je robustan i radi bez grešaka promenama iz jednog stanja u drugo. Može se minimalno refaktorisati kod.

Moguće nadogradnje su dodavanje i semafora za drugu ulicu kako bi simulirali rad raskrasnice.

Takođe je moguće dodavanje kamera u realnim uslovima upotrebe sa implementacijom Mašinskog učenja u detekciji vozila kako bi se produžilo minimalno trajanje zelenog svetla za vozila ako je gužva na putu u zavisnosti od ulice.

Moguće je i dodavane WiFi konekcije kako se udaljeno upravljalo sistemom od strane korisnika.