

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 odnosi 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);** .

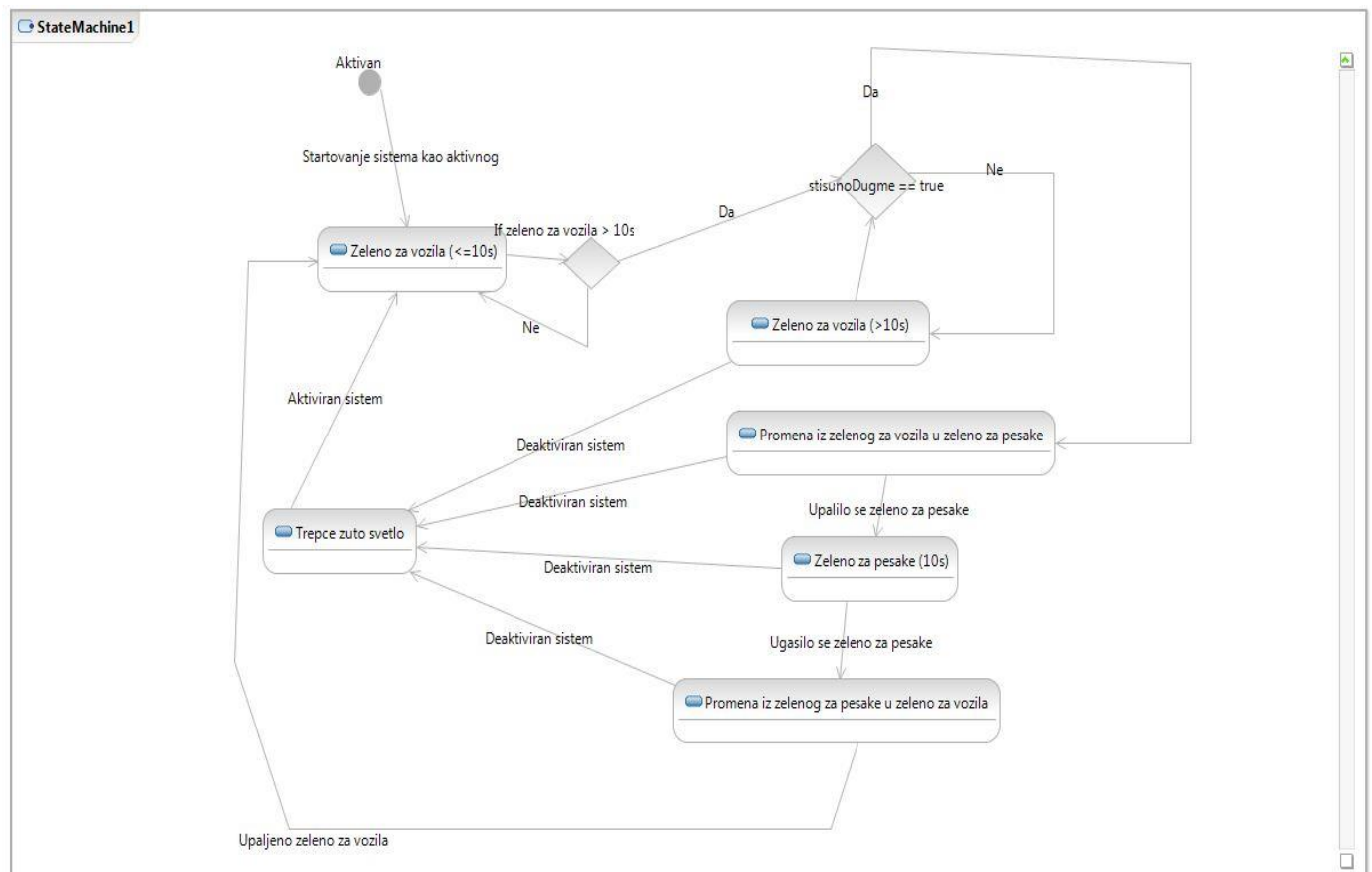
Umesto tajmera je korišćena funkcija **delay(interval);** kojom se Arduino sistem pauzira na određeno vreme. Zato imamo proveru na prvom arduinu nakon svakog **delay()** da li ima promene stanja sistema, kako bi se što pre zaista promenilo stanje na prvom arduinu. Sistem ima 2 glavna stanja (aktivan, 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 funkcioniše 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).

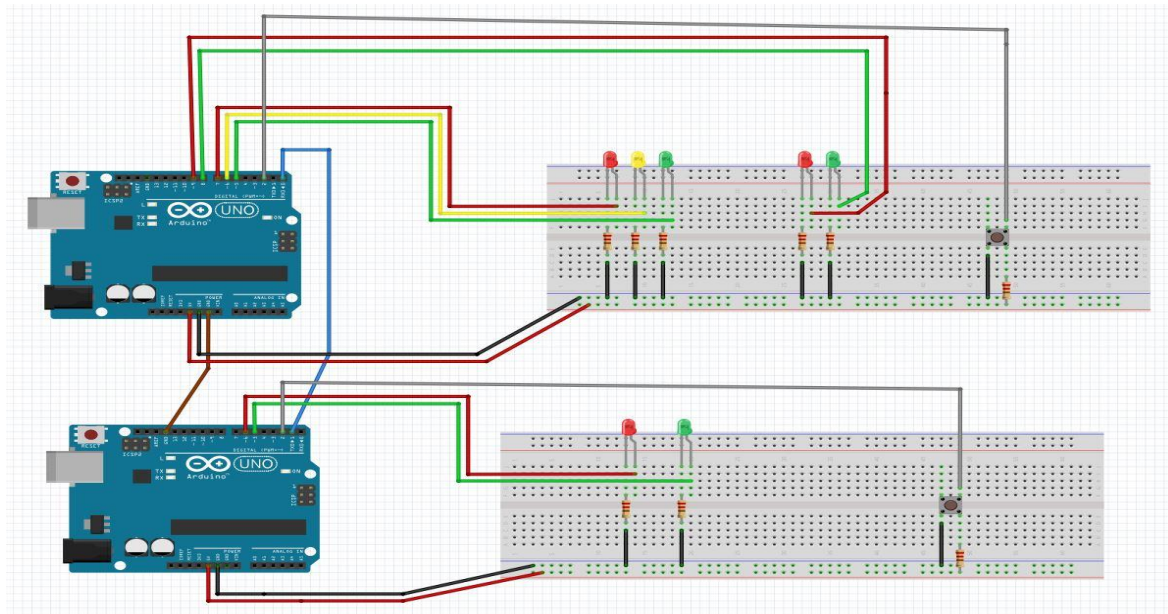
Na početku je sistem aktivan i ulazi u stanje prvih 10 sekundi upaljenog svetla za vozila. Nakon 10 sekundi se proverava da li je dugme stisnuto, ako jeste onda ide sekvenca paljenja i gašenja svetla za pešački prelaz. Ako nije stisnuto, nastavlja se sekvenca zelenog svetla sve dok se ne stisne dugme 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 trepće žuto svetlo, i pri ponovnoj aktivaciji je sistem u stanju prvih 10 sekundi zelenog svetla za vozila.

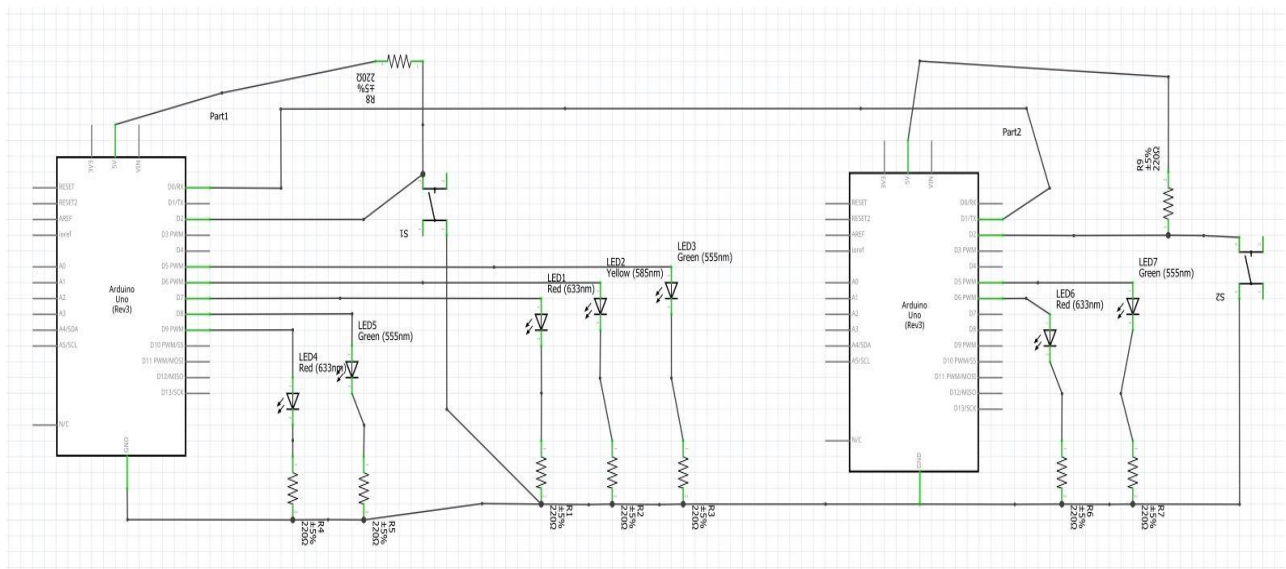


3.Fizička realizacija i šema električnog kola

2 Arduina su povezana jednom žicom za prenos podataka, jer je komunikacija tipa Simplex (Arduinio za (de)aktivaciju sistema je transmitter, dok je Arduino za upravljanje semaforima isključivo prijemnik). Takođe smo povezali oba arduinia 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). Tu je promenljiva **prevMillis** za ažuriranje vremena nakon promena stanja, kao i intervali za svetla semafora i **blink** promenljiva za blinkanje žutog svetla kada je sistem neaktivan.

```
#define VOZILA_CRVENO 7
#define VOZILA_ZUTO 6
#define VOZILA_ZELENO 5
#define PESACI_CRVENO 9
#define PESACI_ZELENO 8
#define PESACI_DUGME 2
bool stisnuoDugme = false;
bool jeAktivanFlag = true;
int blink = HIGH;
unsigned long prevMillis;
unsigned long intervalZeleno = 10000;
unsigned long intervalZuto = 3000;
unsigned long intervalCrvenoVozila = 2000;
unsigned long intervalCrvenoPesaciDoZutogVozila = 1000;
short stanjeSemafor = 0;
```

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.

```
11 // Interrupt dugme pomocna funkcija,
12 // da registruje zahtev pesaka nakon 10 sekundi zelenog za vozila.
13 void pesakStisnuo(){
14     stisnuoDugme = true;
15 }
```

Funkcija koja proverava da li ima promene stanja na serijskom portu (da li ima nove poruke), i samo ako ima onda učitava 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** isključuje žuto svetlo koje je treptalo za neaktivan sistem i uključuje zeleno za vozila i crveno za pešake i takođe resetuje sve flegove na inicijalno stanje. Ako je deaktiviran sistem onda pogasi sa svetla i upali žuto da trepće.

```

void proveriStanje(){
    // Provera da li ima poruke na serijskom portu.
    if(Serial.available() > 0){
        String stanje = Serial.readString();

        // Ako je sistem pocinje sa aktivan (zbog \r\n,
        // setuj stanje na aktivan i inicijalizuj na pocetak semafor vozila.
        // Ugasi zuto svetlo nakon aktivacije sistema.
        if(stanje.startsWith("aktivan")){
            digitalWrite(VOZILA_ZUTO, LOW);
            digitalWrite(VOZILA_ZELENO, HIGH);
            digitalWrite(PESACI_CRVENO, HIGH);
            jeAktivanFlag = true;
            blink = true;
            stanjeSemafor = 0;
            stisnuoDugme = false;
        } else {
            digitalWrite(VOZILA_ZELENO, LOW);
            digitalWrite(VOZILA_CRVENO, LOW);
            digitalWrite(PESACI_ZELENO, LOW);
            digitalWrite(PESACI_CRVENO, LOW);
            digitalWrite(VOZILA_ZUTO, blink);
            jeAktivanFlag = false;
        }
        prevMillis = millis();
    }
}

```

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 **pesakStisnuo**.

```

void setup(){
    // Otpocinje serijska komunikacija, ova vrednost je najniza frekvencija.
    Serial.begin(9600);
    pinMode(VOZILA_CRVENO, OUTPUT);
    pinMode(VOZILA_ZUTO, OUTPUT);
    pinMode(VOZILA_ZELENO, OUTPUT);
    pinMode(PESACI_CRVENO, OUTPUT);
    pinMode(PESACI_ZELENO, OUTPUT);
    pinMode(PESACI_DUGME, INPUT);
    digitalWrite(VOZILA_ZELENO, HIGH);
    digitalWrite(PESACI_CRVENO, HIGH);

    // Prekidna rutina za pritisak pesaka
    attachInterrupt(digitalPinToInterrupt(PESACI_DUGME), pesakStisnuo, RISING);
}

```

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 kod u else bloku na dnu gde su sva svetla pogašena I blinka žuto svetlo na svakih 0.5s . Ako je aktivan sistem onda proveri da li je prošlo 10 sekundi od početka zelenog za vozila I ako jeste onda proveri da li je stisnuto dugme. Ako je stisnuo dugme pešak, onda setuje stanje na 1 I pali žuto za vozila. I posle kako se vrti loop u narednom if će proveravati koje je stanje (stanja se inkrementiraju istim redom kako se menjaju svetla u sekvenci za pešake) I ako je istekao period za to stanje onda promeni svetla I prebaci u sledeće stanje. Nakon isteka poslednjeg stanja (stanje + 5 - žuto I crveno za vozila) sistem se vraća u inicijalno 0 stanje I resetuje promenljivu stisnuoDugme na false.

Ako se desi kojim slučajem da se deaktivira sistem tokom sekvence za pešake, sistem će se u toku sekvence ugasi čim dočeka poruku od transmitera I preći u stanje trepćućeg žutog svetla.

```

void loop(){
    proverStanje();
    if(jeAktivanFlag){
        // Drzi minimum deset sekundi zeleno za vozila.
        // I onda proveriti da li je stisnuto dugme za semafor.
        bool aktiviratiSekvencuZaPesake = stisnutoDugme && stanjeSemafor == 0;
        if(millis() - prevMillis >= intervalZeleno && aktiviratiSekvencuZaPesake){
            digitalWrite(VOZILA_ZELENO, LOW);
            digitalWrite(VOZILA_ZUTO, HIGH);
            stanjeSemafor = 1;
            prevMillis = millis();
        }
        // Ako je promenjeno stanje sa zelenog za vozila za pesake
        if(stanjeSemafor != 0){
            // Prosao je period za zuto za vozila, prebaci u stanje za crveno.
            if(stanjeSemafor == 1 && millis() - prevMillis >= intervalZuto){
                stanjeSemafor = 2;
                digitalWrite(VOZILA_ZUTO, LOW);
                digitalWrite(VOZILA_CRVENO, HIGH);
                prevMillis = millis();
            } else if (stanjeSemafor == 2 && millis() - prevMillis >= intervalCrvenoVozila){
                stanjeSemafor = 3;
                digitalWrite(PESACI_CRVENO, LOW);
                digitalWrite(PESACI_ZELENO, HIGH);
                prevMillis = millis();
            } else if (stanjeSemafor == 3 && millis() - prevMillis >= intervalZeleno){
                stanjeSemafor = 4;
                digitalWrite(PESACI_ZELENO, LOW);
                digitalWrite(PESACI_CRVENO, HIGH);
                prevMillis = millis();
            } else if (stanjeSemafor == 4 && millis() - prevMillis >= intervalCrvenoPesaciDoZutogVozila){
                stanjeSemafor = 5;
                digitalWrite(VOZILA_ZUTO, HIGH);
                prevMillis = millis();
            } else if (stanjeSemafor == 5 && millis() - prevMillis >= intervalZuto){
                stanjeSemafor = 0; // Vrati na zeleno za vozila i crveno za pesake
                digitalWrite(VOZILA_CRVENO, LOW);
                digitalWrite(VOZILA_ZUTO, LOW);
                digitalWrite(VOZILA_ZELENO, HIGH);
                stisnutoDugme = false;
                prevMillis = millis();
            }
        }
        // Deaktiviran sistem i onda samo zuto svetlo blinka.
    } else {
        if(millis() - prevMillis >= 500){
            blink = !blink;
            digitalWrite(VOZILA_ZUTO, blink);
            prevMillis = millis();
        }
    }
}

```

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 kao i prevMillis i blink za blinkanje diode.

```

1 #define ZELENO_AKTIVAN 5
2 #define CRVENO_NEAKTIVAN 6
3 #define AKTIVACIJA_DEAKTIVACIJA_DUGME 2
4 int neaktivan = LOW; // U pocetku je aktivan sistem.
5 unsigned long prevMillis;
6 unsigned long interval = 500; // Interval za blinkanje - 500ms.
7 bool blink = true;
8

```

Funkcija 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.

```

// Interrupt funkcija za promenu stanja i tansmisiju stanja,
// Transmitter salje kao output, dok ce prvi arduino prihvatiti kao Prijemnik.
void promeniStanje(){
    neaktivan = !neaktivan;
    String output = (neaktivan == HIGH) ? "neaktivan" : "aktivan";
    Serial.println(output);
    blink = true;
    prevMillis = millis();
}

```

Na početku je inicijalizovana 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).

```
14 void setup(){
15   Serial.begin(9600); // Inicijalizuj serijsku komunikaciju.
16   pinMode(ZELENO_AKTIVAN, OUTPUT);
17   pinMode(CRVENO_NEAKTIVAN, OUTPUT);
18   pinMode(AKTIVACIJA_DEAKTIVACIJA_DUGME, INPUT);
19   attachInterrupt(digitalPinToInterrupt(AKTIVACIJA_DEAKTIVACIJA_DUGME), promeniStanje, RISING);
20 }
```

U glavnoj petlji u zavisnosti od vrednosti neaktivan koja se invertuje nakon prekida će uvek jedna dioda da treperi (svetli 0.5s i biti ugašena 0.5s), dok će druga dioda biti konstantno ugašena. Kontrolisanje blinkanja se odvija korišćenje interval koji je 0.5s i kad god se prekorači taj interval, invertuje se blink i ažurira se `prevMillis` sve dok se opet ne dosegne interval.

```
void loop(){
  // Kontrolisemo blinkanje upaljene diode
  if(millis() - prevMillis >= interval){
    blink = !blink;
    prevMillis = millis();
  }
  if(blink){
    // Invertuj paljenje i gasenje dioda.
    digitalWrite(CRVENO_NEAKTIVAN, neaktivan);
    digitalWrite(ZELENO_AKTIVAN, !neaktivan);
  } else {
    if (neaktivan == LOW) digitalWrite(ZELENO_AKTIVAN, neaktivan);
    else digitalWrite(CRVENO_NEAKTIVAN, !neaktivan);
  }
}
```

5. Zaključak i moguća nadogradnja

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