

Internet Stvari

Projektna dokumentacija za 2. Kolokvijum

1. Uvod

Cilj projekta je realizacija sistema za upravljanje osvetljenjem uz pomoć 2 Arduino Uno (Rev3) mikrokontrolera, diode, manulenog prekidača, releja, LCD displeja, 2 tastera i senzora temperature i osvetljenja. Jedan arduino služi za senzore i kontrolu svetlja dok drugi predstavlja interfejs ka korisniku.

Čitanje sa manulenog prikadača se čita unutar **loop()** funkcije nakon što se proverí da li je sistem uopšte u manuelnom stanju upavljanja. Uzima se vrednost iz **digitalRead(MANUAL_SWITCH)** funkcije kojoj se kao argument prosleđuje konstanta koja ima vrednost digitalnog pina na koji je povezan mehanički prekidač.

Takođe na drugom Arduino imamo 2 tastera gde se pritiskom na njih izaziva prekid u funkciji koja je definisana u `attachInterrupt()`. Jedan taster služi za promenu stanja sistema iz manualnog u automatsko i obrnuto a drugi taster služi za indirektno paljenje/gašenje diode preko releja kada je u režim rada u automatskom.

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

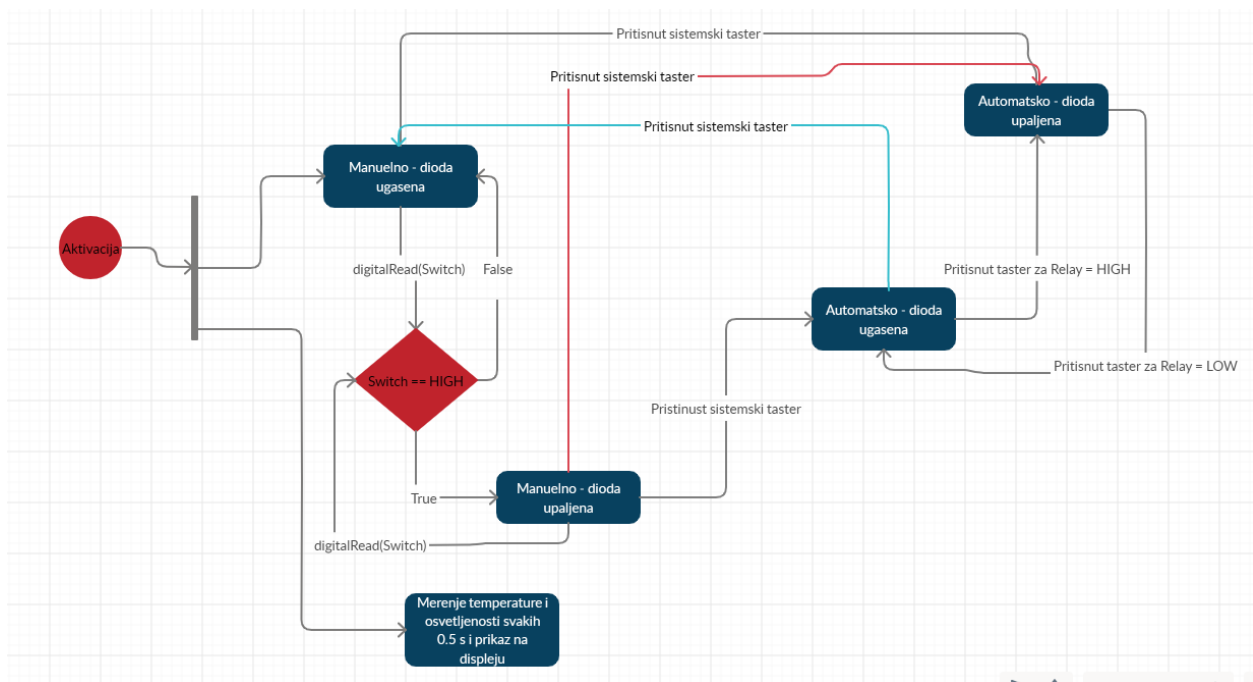
Na oba arduina se u posebnoj funkciji za proveru zahteva proveravala ima li dolazne poruke na serijskom portu za prijem komandom **Serial.available() > 0**. Zatim pomoću funkcije **Serial.readString()** učitavamo string sa izlaza drugog arduina. Drugi arduino šalje na izlaz poruke pomoću naredbe **Serial.println()**.

Imamo 2 glavna stanja (manuelno, automatsko), i unutar svakog od tih stanja imamo podstanja kada je diode upaljena ili ugašena, u međuvremenu se meri temperatura i osvetljenost. Dijagramom stanja je prikazan prelaz iz jednog u drugo stanje pod određenim uslovima.

2. Dijagram stanja

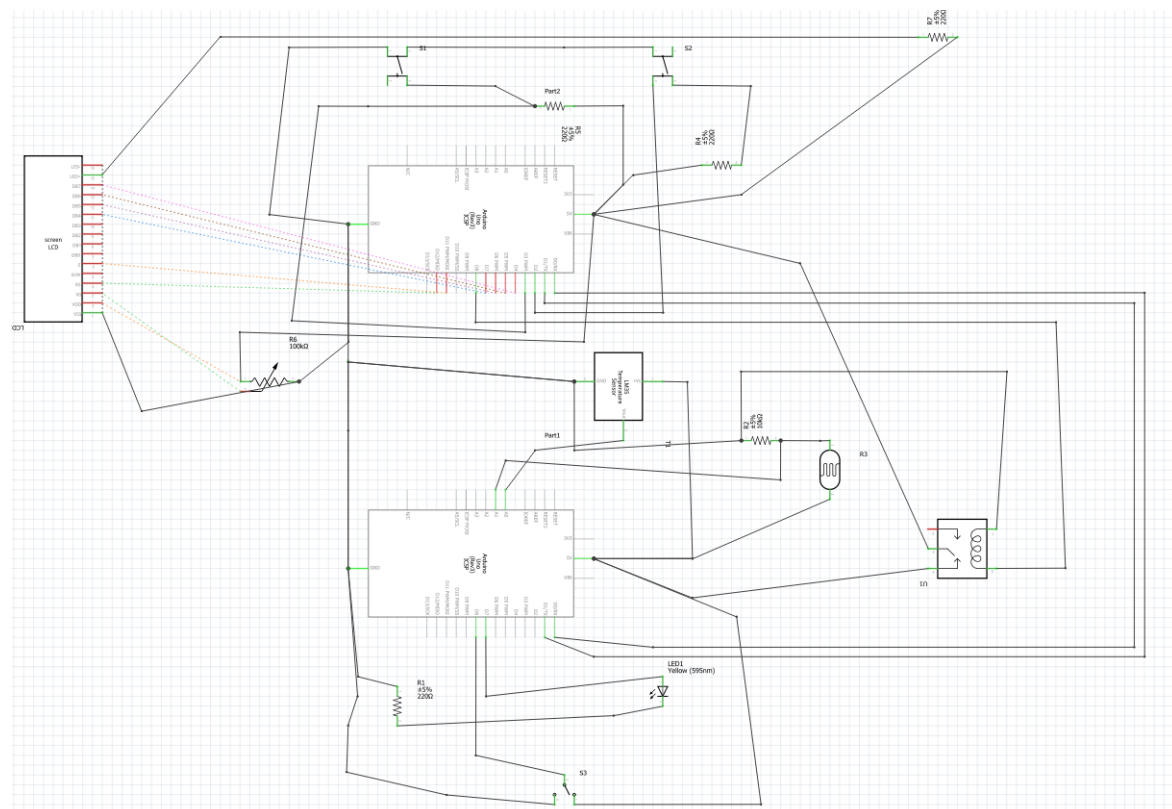
Ceo sistem funkcioniše u 2 glavna stanja kojima upravlja drugi arduino posredstvom sistemskog tastera, za ostala 2 podstanja imamo još jedan taster za automatsko paljenje/gašenje i manualni prekidač na 1. arduinu. Merenje sa senzora je konstantno i nije zavisno ni od jednog stanja ili podstanja, osim kada se prikazuje.

Na početku je sistem u manuelnom režimu rada i može se paliti ili gasiti dioda samo pomoću manualnog prekidača. Promenom u automatsko stanje, blokira se mogućnost daljeg paljenja ili gašenja diode pomoću manualnog prekidača već se to sad može realizovati indirektno preko releja i tastera sa drugog arduina. Pritiskom na sistemski taster, sistem se opet vraća u manualni režim i mehanički prekidač se opet može koristiti za paljenje ili gašenje diode.



3. Fizička realizacija i šema električnog kola

2 Arduina su povezana sa dve žice za prenos podataka, jer je komunikacija tipa Full Duplex. Takođe smo povezali oba arduina preko GROUND pinova da bi izjednačili potencijal na oba mikrokontrolera. Za diode i tastere smo koristili otpornike od 220 oma, dok smo za fotooptornik koristili otpornik od 10 000 oma.



4. Analiza koda programa

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

1.Arduino

Na prvom arduinu su priključeni senzori temperature i osvetljenja, dioda kao i izlazni deo NO pina SPDT releja i mehanički prekidač Slide Switch.

Definisali smo konstante kao brojeve pinova za diodu, bool promenljivu za definisanje sistemskog stanja rada potencijalnog neutralisanja komandi mehaničkog prekidača i period od 500 milisekundi koji će služiti da na svakih 0.5 sekundi merimo fizičke veličine sa senzora. Na kraju je definisana promenljiva za ažuriranje perioda rada arduina nakon svakih 0.5 sekundi.

```
#define DIODE 7
#define MANUAL_SWITCH 8
bool manualState = true;
int measurePeriod = 500;
unsigned long prevMillis;
```

Ovde analogni signal sa fotootpornika prevodimo u lume kao mernu jedinicu osvetljenja. Najpre analogni signal pretvaramo u napon i zatim prema Omovom zakonu napon prevodimo u otpor (koristili smo otpor od 10k oma) i na kraju otpor prevodimo u lume.

```
int toLumens(float raw){
    float Vout = raw * (5 / float(1023)); // Conversion analog to voltage.
    float RLDR = (10000 * (5 - Vout))/Vout; // Conversion voltage to resistance.
    int phys=500/(RLDR/1000); // Conversion resistance to lumen.
    return phys;
}
```

U **setup()** funkciji smo definisali brzinu serijske komunikacije na **baud rate 9600** (bitova po sekundi), izlazni signal za diodu i ulazni od mehaničkog prekidača kao i ulazne analogne signale sa senzora temperature i fotootpornika.

```
void setup(){
    Serial.begin(9600);
    pinMode(DIODE, OUTPUT);
    pinMode(MANUAL_SWITCH, INPUT);
    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
}
```

U glavnoj **loop()** funkciji proveravamo da li je pristigla promena sistemskog stanja na serijskom portu. Ako jeste onda se invertuje sistemsko stanje i priveremeno gasi dioda. Zatim ako je trenutno stanje manuelno onda se učitava vrednost sa mehaničkog prekidača i šalje signal diodi. Zatim se na svakih 0.5 sekundi radi manjeg opterećenja mikrokontrolera učitava vrednost temperature i osvetljenosti po određenim formulama radi konverzije temperature u celzijuse i osvetljenja u lume. Zatim se sa tim vrednostima sklapa string koji se šalje preko serijske veze drugom arduinu za ispis na LCD displeju.

```

void loop(){
    if(Serial.available() > 0){
        manualState = (Serial.readString().startsWith("1")) ? true : false;
        digitalWrite(DIODE, LOW);
    }
    if(manualState){
        int switchState = digitalRead(MANUAL_SWITCH);
        digitalWrite(DIODE, switchState);
    }

    // Measure on every 0.5 seconds.
    if(millis() - prevMillis >= measurePeriod){
        float temp = ((analogRead(A0) * 0.00488) - 0.5) * 100;
        int lighting = toLumens(analogRead(A1));
        String tempLighting = ((String) temp) + "C" + ((String)lighting) + "L";
        Serial.println(tempLighting); // Temp in C and lighting in lumens.
        prevMillis = millis();
    }
}

```

2.Arduino

Na drugom arduinu se nalazi interfejs ka korniku (LCD displej sa potencijometrom za contrast i 2 tastera od kojih je jedan softverski povezan sa ulazom releja.

Uvezli smo biblioteku za LCD displej i instancirali ga kao objekat sa 12 i 11 Ready State i Enable pinovima i 7, 6, 5, 4 digitalnim pinovima. Takođe smo definisali kao konstante pinove za oba tastera i SPDT relej i stanje koje se kao signal šalje ka releju kao izlazni signal, kao i sistemsko stanje da bi smo imali sinhronizaciju sa prvim arduinom.

```

#include <LiquidCrystal.h>
#define BUTTON_SYSTEM_STATE 2
#define BUTTON_AUTO_LED_STATE 3
#define RELAY_PIN 8
bool manualState = true;
int relayToDiodeState = LOW;
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);

```

Funkcija koja uzima kao paramtere 2 stringa koji će biti uneti u 2 reda LCD displeja. Najpre se obrišu prethodni karakteri ako ih je bilo i unesu se red po red vrednosti iz parametara.

```

void lcdPrint(String row1, String row2){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(row1);
    lcd.setCursor(0, 1);
    lcd.print(row2);
}

```

Funkcija koja se okida na prekid klika sistemskog dugmeta gde se invertuje sistemsko stanje. Ako je promena na manuлено stanje onda setuj nizak naponski nivo ka releju i prikaži poruku na LCD displeju o promeni stanja u oba slučaja. Na kraju pošalji serijskom vezom promenu stanja na prvi arduino radi sinhronizacije.

```

void changeSystemState(){
    manualState = !manualState;
    if(manualState){
        relayToDiodeState = LOW;
        digitalWrite(RELAY_PIN, relayToDiodeState);
        lcdPrint("Aktiviran je", "manulessly rezim");
    } else {
        lcdPrint("Aktiviran je", "automatski rezim");
    }
    Serial.println(manualState);
}

```

Funkcija koja se okida na prekid tastera za automatsko paljenje i gašenje diode preko releja. Ako je stanje manuelno onda samo prikaži poruku da je manuelni režim rada što znači da je taster onemogućen da automatski aktivira relej koji pali/gasi diodu. U suprotnom invertuj stanje releja i prikaži poruku na LCD displeju u prvom redu u zavisnosti od toga da li je diode upaljena ili ugašena.

```
void changeDiodeState(){
  if(manualState){
    lcdPrint("Manuleni je", "rezim rada !");
  } else {
    relayToDiodeState = !relayToDiodeState;
    digitalWrite(RELAY_PIN, relayToDiodeState);
    String msg = (relayToDiodeState == HIGH) ? "Dioda upaljena !" : "Dioda ugasena !";
    lcdPrint(msg, "");
  }
}
```

U **setup()** funkciji smo setovali serijsku brzinu slanja podataka na **9600 bita po sekundi** kao baud rate i paljenje LCD ekrana sa 16 kolona i 2 vrste. Takodje smo setovali ulazni signal sa 2 tastera i izlazni prema releju. Za 2 tastera smo definisali prekide sa **attachInterrupt()** funkcijom na definisanim digitalnim pinovima, okidačke funkcije i mod **RISING** (što znači da se prekid okida kada se sa LOW primi HIGH signal jer su tasteri u **PULL UP** modu i otpornik od 220 oma je na naponu od 5V.

```
void setup(){
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(BUTTON_SYSTEM_STATE, INPUT);
  pinMode(BUTTON_AUTO_LED_STATE, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(BUTTON_SYSTEM_STATE), changeSystemState, RISING);
  attachInterrupt(digitalPinToInterrupt(BUTTON_AUTO_LED_STATE), changeDiodeState, RISING);
}
```

U glavnoj loop() funkciji samo proveravamo da li je vrednost dostupnih bita na serijskom portu veća od nule i ako ima pristiglih podataka sa prvog arduina, onda su to podaci sa senzora koje delimo sa delimiterima, izvlačimo vrednosti i pretvaramo u adekvatan string za ispis na LCD ekranu.

```
void loop() {
  if(Serial.available() > 0){
    String tempLight = Serial.readString();
    int delim1 = tempLight.indexOf("C");
    int delim2 = tempLight.indexOf("L");
    String temp = "Temp: " + tempLight.substring(0, delim1) + " C";
    String light = "Light: " + tempLight.substring(delim1+1, delim2) + " lm";
    lcdPrint(temp, light);
  }
}
```

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 da se diode pali padom osvetljenja I da se doda ventilacija ili grejač koji bi se palili/gasili u zavisnosti od promene temperature. To bi bio dobar način za iskorišćenje senzora. Takođe bi se sistemu mogao ugraditi Wi-fi modul i time omogućilo da korisnik preko posebne aplikacije kontroliše osvetljenje efikasno.