

Izveštaj projektnog zadatka

Student: Milan Savić

Broj indeksa: 2016203834

U nastavku je dat kod implementacije prvog Arduino uređaja koji predstavlja simulaciju kuće u kojoj se nalazi dioda, senzor temperature i senzor osvetljenja.

```
#define SensorTempPin A0
#define ResistorPin A1
const int SwitchPinNo = 12;
const int RelayPinNo = 13;

String poruka;
String zaSlanje;

unsigned long pocetno;
unsigned long trenutno;
unsigned long vreme;
unsigned long interval = 1000; // na 1000ms Arduino1 salje podatke Arduino2

int temp;
int osvetljenje;

void setup()
{
    Serial.begin(9600);
    pinMode(SwitchPinNo, OUTPUT);
    pinMode(RelayPinNo, OUTPUT);
    digitalWrite(SwitchPinNo, HIGH);
    digitalWrite(RelayPinNo, LOW);
    pocetno = millis();
}

void procitajPoruku() {
    poruka = Serial.readString();
    if (poruka == "m\r\n") {
        digitalWrite(SwitchPinNo, HIGH);
        digitalWrite(RelayPinNo, LOW);
    } else {
        digitalWrite(SwitchPinNo, LOW);
        if (poruka == "upali\r\n") {
            digitalWrite(RelayPinNo, HIGH);
        } else if (poruka == "ugasi\r\n") {
            digitalWrite(RelayPinNo, LOW);
        }
    }
}

void ocitajIPosalji() {
    temp = analogRead(SensorTempPin);
    temp = (temp * 0.00488 - 0.5) * 100;
    osvetljenje = analogRead(ResistorPin);
    zaSlanje = String(temp) + " " + String(osvetljenje) + "k";
    Serial.print(zaSlanje);
    zaSlanje = "";
}

void loop()
{

```

```

if(Serial.available()>0){
    procitajPoruku();
}
trenutno = millis();
if((trenutno-pocetno)>interval){
    pocetno = trenutno;
    ocitajIPosalji();
}
}

```

Na početku se definišu konstante koje određuju na kom pinu su povezani senzor temperature, fotootpornik, prekidač i relej.

U metodi setup() definiše se brzina komunikacije između dva Arduino uređaja, definiše se pinMode prekidača i releja, upisuju se stanja na prekidači i releju. U početnom stanju prekidač je na HIGH, a relej LOW. Zbog nemogućnosti korišćenja timer-a u Tinkercad-u, pomoću funkcije millis se simulira rad timera. Pri pokretanju očitava se početno vreme koje će se kasnije upoređivati sa trenutnim i na taj način odrediti interval očitavanja podataka sa senzora.

U metodi loop() proverava se da li postoji poruka i da li je vreme za očitavanje podataka sa senzora i slanje drugom uređaju.

Ukoliko postoji poruka ona se očitava i u zavisnosti od sadržaja uključuje se manuelan ili automatski režim rada. U manuelnom režimu rada stanje na pinu za relej je LOW, a za prekidač HIGH. U automatskom režimu rada stanje za prekidač je LOW. U automatskom režimu rada u zavisnosti od vrednosti poruke, dioda se može paliti i gasiti menjanjem stanja na pinu za relej.

Kada prođe vreme od 1000ms čitaju se podaci sa senzora i šalju se drugom Arduino. Poruka koja se šalje je u formatu *temperatura razmak osvetljenje* i slovo *k* koje označava kraj poruke.

U nastavku je dat kod implementacije drugog Arduino uređaja koji kontroliše rad prvog uređaja i prikazuje korisniku informacije na LCD displeju.

```

#include <LiquidCrystal.h>

```

```

const int DispHeight = 16;
const int DispWidth = 2;

```

```

const int RSPinNo = 12;
const int EnablePinNo = 11;
const int D4PinNo = 4;
const int D5PinNo = 5;
const int D6PinNo = 6;
const int D7PinNo = 7;

```

```

const int Button1PinNo = 3;
const int Button2PinNo = 2;

```

```

bool manuelanRezimRada = true;
bool diodaUpaljena = false;

```

```

String poruka;

LiquidCrystal LCD(RSPinNo,EnablePinNo,D4PinNo,D5PinNo,D6PinNo,D7PinNo);

void promeniRezimRada(){
    manuelanRezimRada = !manuelanRezimRada;
    if(manuelanRezimRada){
        diodaUpaljena = false;
        Serial.println("m");
        stampajNaLcdUDvaReda("Manuelan","rezim rada");
    }else{
        Serial.println("a");
        stampajNaLcdUDvaReda("Automatski","rezim rada");
    }
}

void promeniStanjeDiode(){
    if(!manuelanRezimRada){
        diodaUpaljena = !diodaUpaljena;
        if(diodaUpaljena){
            Serial.println("upali");
            stampajNaLcd("Dioda upaljena");
        }else{
            Serial.println("ugasi");
            stampajNaLcd("Dioda ugasena");
        }
    }else{
        stampajNaLcdUDvaReda("Manuelan","rezim rada");
    }
}

void stampajNaLcd(String poruka){
    LCD.clear();
    LCD.setCursor(0,0);
    LCD.print(poruka);
}

void stampajNaLcdUDvaReda(String poruka1, String poruka2){
    LCD.clear();
    LCD.setCursor(0,0);
    LCD.print(poruka1);
    LCD.setCursor(0,1);
    LCD.print(poruka2);
}

void setup()
{
    Serial.begin(9600);
    LCD.begin(DispHeight,DispWidth);
    pinMode(Button1PinNo,INPUT);
    pinMode(Button2PinNo,INPUT);
    attachInterrupt(digitalPinToInterrupt(Button1PinNo), promeniRezimRada,
FALLING);
    attachInterrupt(digitalPinToInterrupt(Button2PinNo), promeniStanjeDiode,
FALLING);
}

```

```

void procitajPoruku() {
    poruka = Serial.readString();
    String temp = "";
    String osvetljenje = "";
    int i = 0;
    while(poruka[i]!=' '){
        temp += poruka[i];
        i++;
    }
    i++;
    while(poruka[i]!='k'){
        osvetljenje += poruka[i];
        i++;
    }
    stampajNaLcdUDvaReda("Temp: "+temp+"C", "Osv:  " + osvetljenje +"Lm");
}

void loop()
{
    if(Serial.available()>0){
        procitajPoruku();
    }
}

```

Na početku se uključuje biblioteka za rad sa LCD displejem i definišu pinovi za displej i tastere.

U metodi setup() definiše se brzina komunikacije između dva Arduino uređaja, definišu se dimenzije LCD displeja, pinMode tastera i dva prekida koja se detektuju prilikom pritiska tastera.

Ukoliko je pritisnut *taster1* menja se režim rada. U zavisnosti od režima rada drugom Arduino se šalje odgovarajuća poruka i vrši se ispis na displeju.

Ukoliko je pritisnut *taster2* menja se stanje diode. Ukoliko je automatski režim rada drugom Arduino se šalje odgovarajuća poruka i vrši se ispis na displeju u suprotnom na displeju se prikazuje da je manuelan režim rada.

U metodi loop() proverava se da li je drugi Arduino uređaj poslao poruku. Ako postoji poruka ona se čita, zatim se parsira i prikazuje se na displeju.