Digitalna soba

Kovačić, Ivan

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:405981

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

dabar

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

DIGITALNA SOBA

Završni rad

Ivan Kovačić

Osijek, 2016.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 24.09.2016.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Ivan Kovačić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3411, 03.09.2012.
OIB studenta:	32431603590
Mentor:	Doc.dr.sc. Ivan Aleksi
Sumentor:	
Naslov završnog rada:	Digitalna soba
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 Postignuti rezultati u odnosu na složenost zadatka: 1 Jasnoća pismenog izražavanja: 3 Razina samostalnosti: 2
Datum prijedloga ocjene mentora:	24.09.2016.
Datum potvrde ocjene Odbora:	28.09.2016.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 29.09.2016.

Ime i prezime studenta:	Ivan Kovačić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3411, 03.09.2012.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: Digitalna soba**

izrađen pod vodstvom mentora Doc.dr.sc. Ivan Aleksi

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

1. UVOD 4
1.1 ZADATAK ZAVRŠNOG RADA 4
2. KORIŠTENE TEHNOLOGIJE 5
2.1 ARDUINO RAZVOJNO OKRUŽENJE 5
2.2 ARDUINO RAZVOJNA MAKETA 5
2.3 ETHERNET SKLOP 7
2.4 RELEJ SKLOP 8
2.5 SENZOR POKRETA HC-SR501 10
3. ELEKTRONIČKI SKLOP 12
3.1 ELEKTRIČNA SHEMA I POPIS KOMPONENTI 12
3.2 SKLOPOVSKI DIO SUSTAVA 14
3.3 PROGRAMSKI DIO SUSTAVA 16
4. ZAKLJUČAK 19
LITERATURA 20
SAŽETAK 21
ABSTRACT 22
ŽIVOTOPIS 23
PRILOG A: Program za Arduino Uno razvojnu maketu 24
PRILOG B: Program za internet stranicu 26

1. UVOD

U ovom radu se obrađuje tema pod nazivom *„Digitalna soba“*. Drugim riječima, potrebno je napraviti programski i sklopovski dio za model digitalne sobe uz pomoć Arduino platforme. Upravljanje Arduino razvojnim sustavom će se odvijati putem jednostavne web stranice koja će biti osposobljena pomoću Ethernet modula i usmjerivača. Cilj je napraviti mikroupravljački sustav koji će regulirati rad uređaja u sobi pomoću senzora pokreta ili putem web stranice. U radu se opisuje Arduino razvojno okruženje što podrazumijeva opis Arduino Uno razvojne makete, arduino razvojnog sustava u kojemu se izvodi programski dio rada, te se opisuje kako se u Arduino okruženju ponašaju razni moduli i senzori.

1.1 ZADATAK ZAVRŠNOG RADA

U ovom završnom radu potrebno je napraviti mikroupravljački sustav koji pomoću raznih senzora i aktuatora obavlja određene radnje u sobi. Neke od njih su otvaranje i zatvaranje vrata, paljenje gašenje svjetla, internet baza podataka s navikama korisnika, udaljeno upravljanje uređajima i sl.

2. KORIŠTENE TEHNOLOGIJE

2.1 ARDUINO RAZVOJNO OKRUŽENJE

Arduino razvojno okruženje sastoji se od tekstualnog urednika za pisanje kôda, prostora za poruke, tekstualne konzole, alatne trake sa tipkama za zajedničke funkcije i od niza izbornika. Povezuje se sa arduidnim sklopovljem kako bi učitao programski kôd i kako bi komunicirao sa Arduinom u realnom vremenu. Programski kod se piše pomoću tekstualnog urednika, te se pohranjuje pod datotečnim nastavkom .ino. Takav programski kod napisan pomoću Arduino razvojnog okruženja zove se skica *(eng. sketch).* Tekstualni urednik podržava pretraživanje i zamjenu teksta, te također i kopiranje odnosno lijepljenje teksta. U doljnjem desnom kutku nalazi se podatak o trenutno spojenoj Arduino razvojnoj maketi, te serijski priključak na koji je isti spojen.

Programi su pisani u programskom jeziku C ili C++. Svaki arduino program mora sadržavati dvije funkcije, a to su *setup() i loop().* Funkcija *setup()* pokreće se jednom na početku programa, te se u njoj definiraju osnovne postavke sustava. Funkcija *loop()* je funkcija koja se cijelo vrijeme poziva dokle god razvojni sustav spojen na napajanje.

Prilikom prenošenja skice na razvojnu maketu koristi se pokretač operacijskog sustava, tj. Program unutar same makete koji omogućava učitavanje podataka na maketu bez potrebe korištenja dodatnog sklopovlja. Pokretač operacijskog sustava bude aktivan nekoliko sekundi dok se maketa ponovno pokreće, nakon čega pokreće novu skicu koja je učitana na razvojnu maketu.

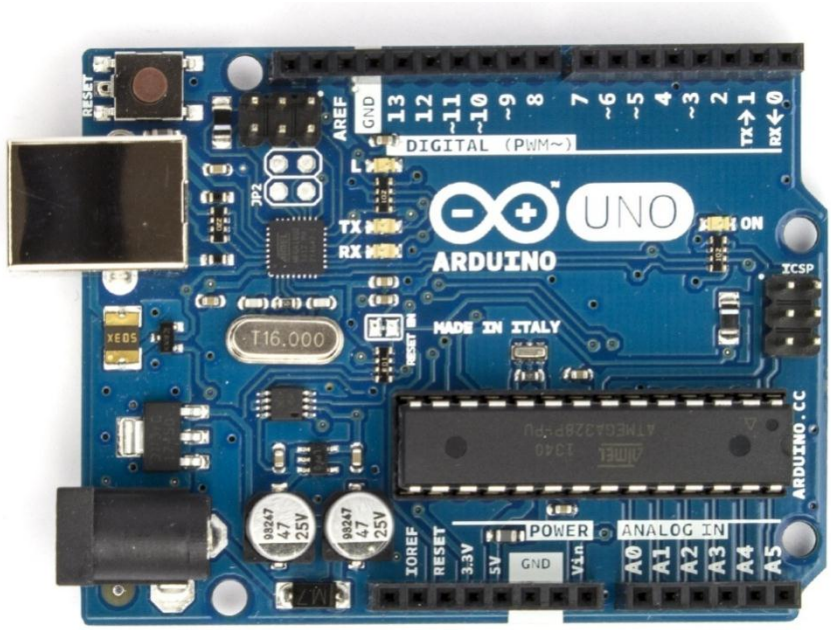
Programske knjižnice pružaju dodatnu funkcionalnost prilikom manipuliranja podataka ili korištenja određenog sklopovlja upravljanog razvojnom maketom.

2.2 ARDUINO RAZVOJNA MAKETA

Arduino UNO razvojna maketa bazirana je na ATmega328P mikroupravljaču. Sastoji se od 14 ulazno/izlaznih priključaka (od kojih 6 podržava modulaciju širine impulsa (eng. PWM)), 6 analognih ulaza, 16MHz keramičkog rezonatora, priključka za USB kao i priključka za napajanje i gumba za ponovno pokretanje.

Arduino razvojna maketa može se programirati putem Arduino razvojnog okruženja. Na ATmega328 postavljen je pokretač operacijskog sustava koji omogućuje implementaciju novog koda bez potrebe za korištenjem dodatnog sklopovlja. Također može se zaobići pokretač operacijskog sustava kako bi se izravno programirao mikroupravljač preko ICSP zaglavlja koristeći arduino programsko sučelje. Arduino UNO ima mogućnost programskog resetiranja bez potrebe fizičkog pritiskanja tipke za reset.

Napajanje može biti dovedeno putem USB priključka ili preko drugog vanjskog izvora. Preporučena vrijednost napajanja iznosi od 7-12V. ATmega328 mikroupravljač sadrži 32KB memorije, uz 0.5KB koju koristi pokretač operacijskog sustava. Sadrži i 2KB SRAM memorije, te 1 KB EEPROM memorije koja može biti pročitana i napisana uz pomoć EEPROM programske knjižnice.



Sl. 2.1. Arduino UNO

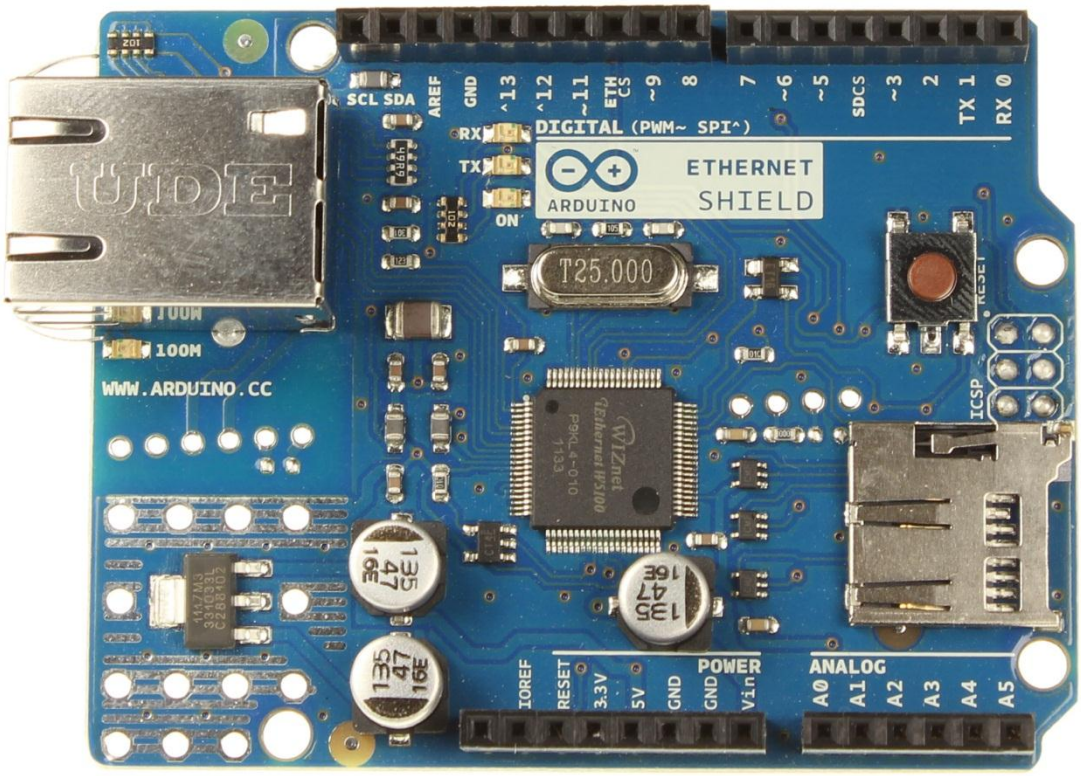
Svaki od 14 digitalnih priključaka na Arduino UNO može se koristiti kao ulaz ili izlaz, koristeći funkcije *pinMode(), digitalWrite() I digitalRead().* Svi priključci rade na naponu od 5V, može osigurati struju do 40mA, te ima unutarnji *pull-up* otportnik od 20-50kΩ. Neki od priključaka imaju posebne funkcije poput priključaka 0 (RX) i 1 (TX), koji mogu primate i slati serijske TTL podatke.

Uz digitalne, Arduino uno sadrži I šest analognih priključaka, označenih od A0 do A6, od kojih svaki podržava 10bit-nu razluživost, odnosno može raspoznati 1024 različite analogne vrijednosti. Po početnim postavkama mogu izmjeriti napone od 0 do 5V, ali moguće je promijeniti gornju granicu koristeći AREF priključak i *analogReference()* funkciju.

Arduino UNO ima vise mogućnosti za komunikaciju s računalom, drugom UNO razvojnom maketom ili drugim mikrokontrolerima. ATmega328 podržava UART TTL serijsku komunikaciju, omogućeno putem digitalnih priključaka (0) RX i (1) TX. Integrirani ATmega16U2 na pločici usmjerava tu serijsku komunikaciju putem USB priključka, te se predstavlja programskom sučelju na računalu kao virtualni *com* priključak. Ardiuno razvojno okruženje na računalu sadrži serijski monitor koji omogućuje slanje i primanje jednostavnih tekstualnih podataka između razvojne makete i računala. Dok se podatci izmjenjuju putem USB sučelja, na maketi će treptati svjetleće RX i TX diode, ali neće prilikom prijenosa na priključcima 0 (RX) i 1 (TX).

2.3 ETHERNET SKLOP

Ethernet shield omogućuje povezivanje Arduina na internet u par koraka. Temeljen je na Wiznet W5100 ethernet čipu koji pruža mrežni stog s TCP i UDP protokolom te podržava do 4 istovremene konekcije. Za korištenje samog sklopa potrebna je Ethernet programska knjižnica koja omogućuje povezivanje na internet. Ethernet razvojna pločica je vrlo slična Arduino razvojnoj pločici, te omogućuje jednostavno povezivanje tako da se umetne s gornje strane Arduino pločice. Ethernet pločica ima standardnu RJ-45 vezu s integriranim linijskim transformatorom i mogućnost napajanja putem Ethernet kabla. Na pločici se također nalazi i micro-SD ulaz, koji se može koristiti za pohranu podataka koji će biti korišteni na mreži. Za korištenje SD čitača potrebna je SD programska knjižnica



Sl. 2.2. Ethernet modul

Arduino komunicira s W5100 sklopom i SD karticom kroz ICSP zaglavlje. Taj proces se odvija na digitalnim pinovima 10, 11, 12 i 13 na Arduino Uno razvojnoj maketi. Na obje pločice priključak 10 se koristi za komunikaciju s W5100 sklopom i priključkom 4 za SD karticu. Ti pinovi ne mogu biti korišteni za generalni ulaz/izlaz.

2.4 RELEJ SKLOP

Relej je zapravo elektronički upravljana sklopka, koji je sposoban upravljati visokim naponima za razliku od Arduina.

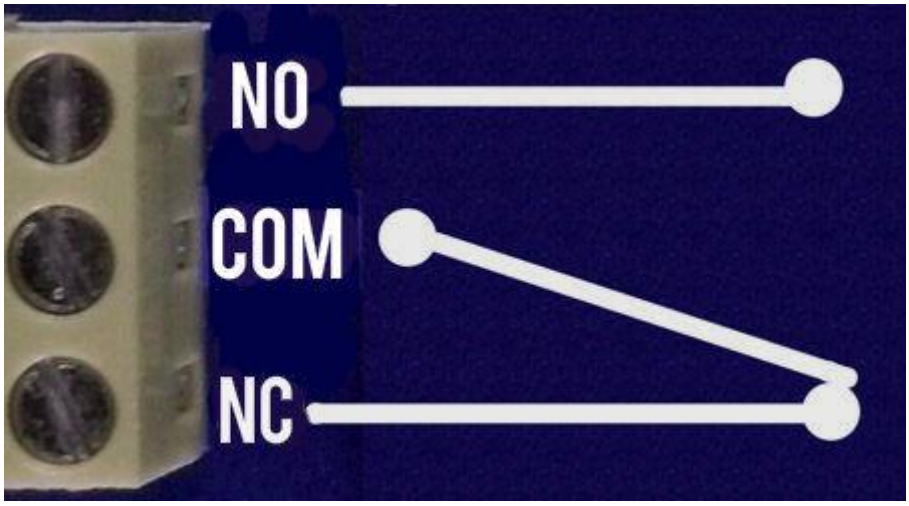
Sl. 2.3. Relej sklop

Relej modul može raditi na manjim te većim naponima, ukoliko je potreban za upravljanje diodama, napajanje mu se može dovesti s arduino pločice, a ukoliko su potrebni veći naponi za upravljanje žaruljama onda mora imati zaseban izvor napajanja od 220V. U primjeru sa slike 2.3. priključci GND i VCC služe za napajanje, dok IN1 priključak služe za upravljanje sklopkom. Kanal ima 3 izlaza, a oni su NC, COM, te NO gdje je:

- NC – *normally closed* (normalno zatvoren)

- Uvijek je spojen s COM, čak i kada je sklop isključen, kada mu se dovede napon, otvori se strujni krug te se prekine veza

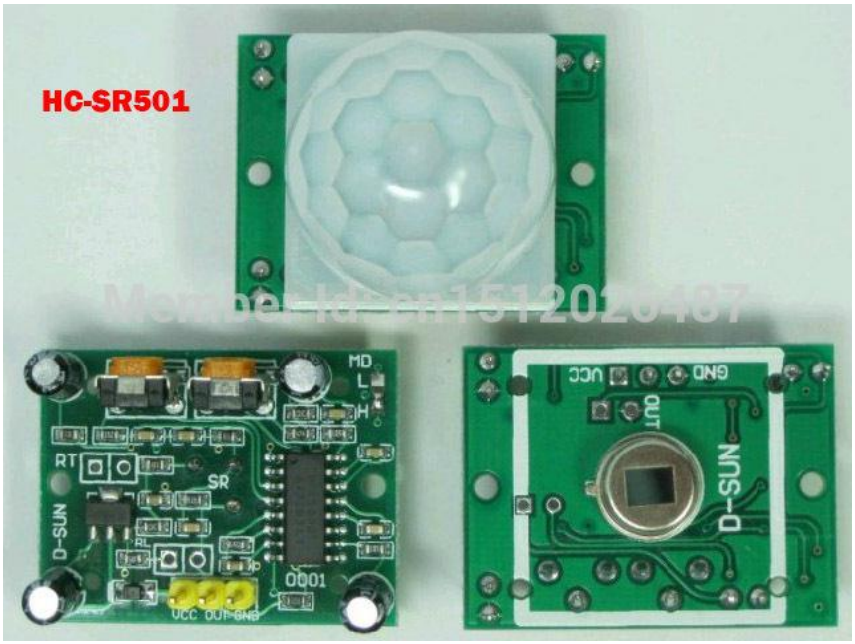
- COM – *Common connection* (zajednička komunikacija)
- NO – *normally open* (normalno otvoren)
    - Ponaša se kao prekidač, s obzirom da je otvoren, nema kontakta s COM. Kada aktiviramo sklop, elektromagnet unutar releja poveže NO i COM te se strujni krug zatvori.



Sl. 2.4. Relej izlazi kanal

2.5 SENZOR POKRETA HC-SR501

HC-SR501 senzor je temeljen na infracrvenoj tehnologiji uz pomoć koje otkriva pokret u svojoj okolini. Lako se spaja na arduino jer ima samo 3 priključka, 2 naponska, te 1 izlazni priključak putem kojeg signalizira arduinu ukoliko primjeti pokret.

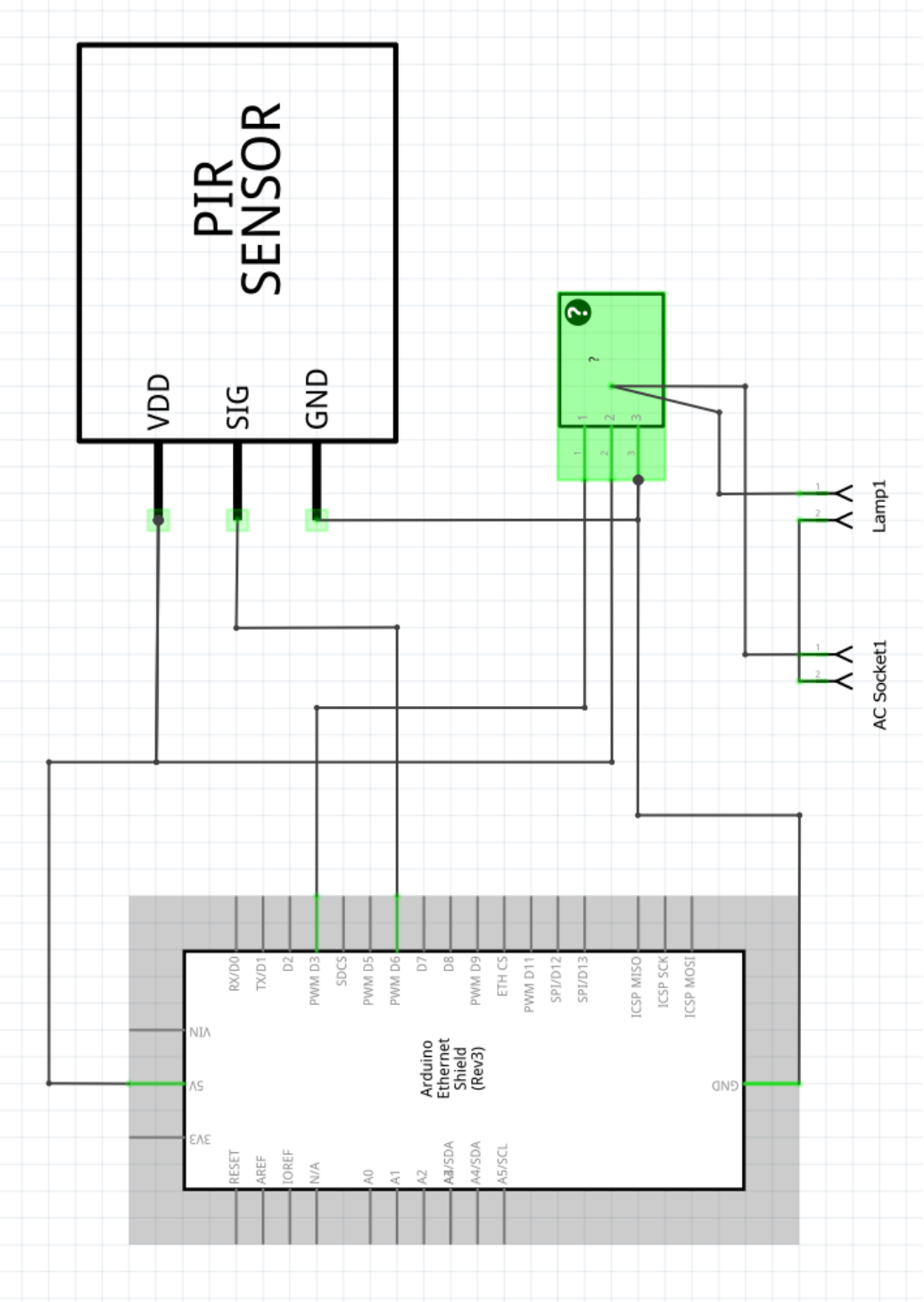


Sl. 2.5. HC-SR501 senzor pokreta

Nakon što se zapazi pokret, izlaz digitalnog priključka se postavlja na HIGH(3.3V). Domet senzora je do 7 metara, a kut kojeg isti pokriva je do 120°. Na senzoru se nalaze još 2 potenciometra od kojih jedan služi za određivanje osjetljivosti na kojoj će senzor opažati pokret, te drugi koji je zadužen za *delay*, odnosno odgodu rada senzora nakon što je pokret detektiran, koji može biti postavljen od 3 sekunde do 5 minuta. Na samome sklopu može se upravljati prekidačem koji možemo postaviti u ponavljajući ili ne ponavljajući način rada. Dok je u ponavljajućem načinu rada senzor će nastaviti davati izlaz sve dok on očitava pokret, dok se u ne ponavljajućem senzor uključi nakon prvog očitanja pokreta, a zatim slijedi se gasi nakon vremena kojeg namjestimo na potenciometru.

3. ELEKTRONIČKI SKLOP

3.1 ELEKTRIČNA SHEMA I POPIS KOMPONENTI



Sl. 3.1. Električna shema sklopa

Na električnoj shemi vidljivo je da su sve komponente povezane na Ethernet maketi, koja je umetnuta na vrh Arduino makete na način da su povezane sve priključnice. Senzor pokreta povezan je na Ethernet priključak 6 koji je postavljen kao izlaz, dok je relej povezan na Ethernet priključak 3 koji je postavljen kao ulaz. Također je vidljivo da je relej povezan putem NO(Normally Open) što znači da prilikom paljenja uređaja mu je potrebno dati vrijednost *0*(*LOW)* kako bi se uređaj upalio, odnosno vrijednost *1(HIGH)* kako bi se uređaj ugasio.

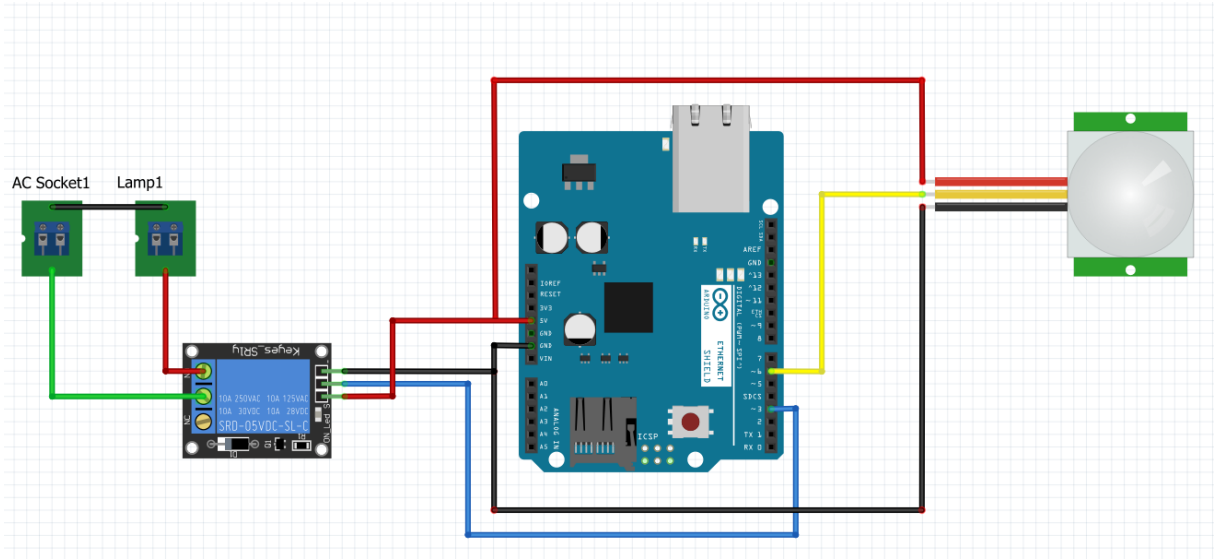
Popis komponenti:

- Arduino UNO Rev3
- Arduino Ethernet Shield R3
- Senzor pokreta HC-SR501
- 1-kanalni relej
- Žarulja (220V)

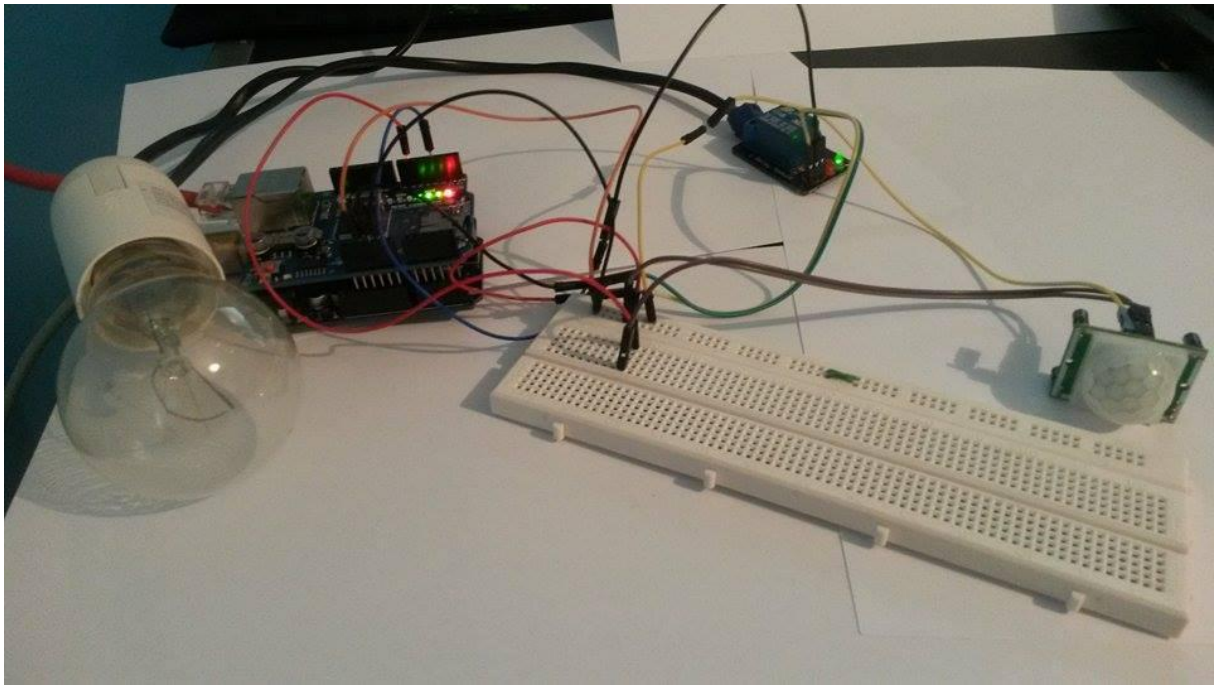
3.2 SKLOPOVSKI DIO SUSTAVA

Sustav je spojen tako da su priključci VCC i GND releja spojeni na priključke 5V i GND Arduino makete, a priključak IN1 je spojen na Arduinov digitalni priključak broj 3 putem kojeg će se moći upravljati sklopkom. Zatim priključak NO s releja priključimo na '+' potrošača, u ovom slučaju sijalice, a '-' sijalice povežemo s jednom žicom od kabla za napajanje. Druga žica tog istog kabla spaja se na COM priključak releja. Nakon što je relej spojen potrebno je spojiti i senzor pokreta, a spaja se na način da priključnice VCC i GND također idu na priključke 5V i GND na Arduino maketi, a signal priključak spaja se na digitalni priključak 6 na Arduino razvojnoj maketi.

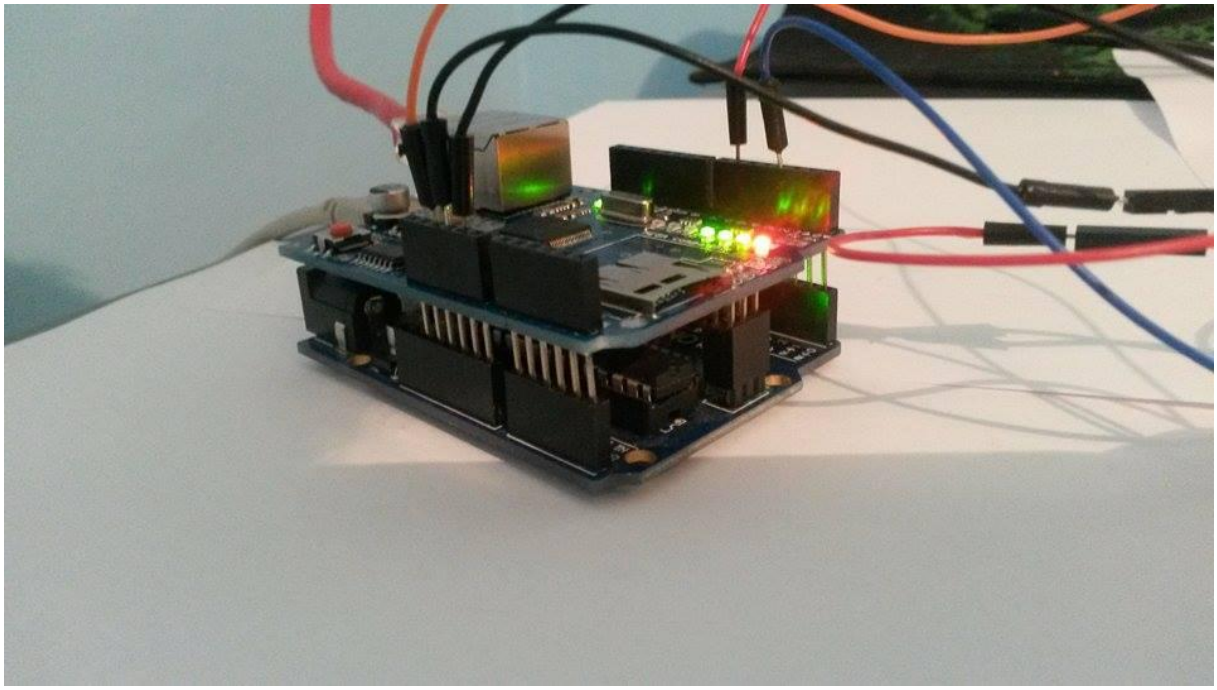
Prilikom uključivanja Arduino makete uključuje se i Ethernet maketa koja pokreće lokalni web server koji služi za upravljanje uređajem, u ovom slučaju svjetlom. Nakon ulaska u web stranicu korisniku se pojave 3 radio tipke od kojih 2 služe za paljenje, odnosno gašenje svjetla, a treća tipka služi za pokretanje senzora pokreta koji preuzima upravljanje uređajem sve dok se to ne izmjeniputem tipki na web stranici.



Sl. 3.2. Model sučelja



Sl. 3.3. Fotografija sustava



Sl. 3.4. Fotografija spoja Ethernet shielda i Arduino makete

Sl. 3.5. Slika upravljačkog dijela sustava

3.3 PROGRAMSKI DIO SUSTAVA

Programska logika ovog sustava opisana je kao formular koji se popunjava sljedeći na način:

```
cl.println("<input type=\"radio\" name=\"relay\" value=\"1\">On<input
type=\"radio\" name=\"relay\" value=\"0\">Off<input type=\"radio\"
name=\"relay\" value=\"2\">PIR</form>");
```

Postavi se ime forme „relay“ koja će kasnije poslužiti kao varijabla koja će pamtiti vrijednost pritisnute tipke, a može poprimit vrijednost 0,1 i 2. Prilikom pritiska na tipku „Submit“ ime forme i njezina vrijednost šalju se klijentu pomoću metode *get.*

```
client.println("<form method=\"get\">");
client.println("<input type=\"submit\" value=\"Submit\">");
```

Potrebno je preuzeti varijablu iz klijenta kako bi se mogla prosijediti na Arduino radi daljnjeg upravljanja sustavom. Preuzimanje varijable vrši se na način da se postavlja nova varijabla *c* tipa *char* koji će pomoću naredbe *client.read();* isčitavati po 1 byte iz klijenta, te ga pohranjivati u *string* iz kojeg će se kasnije moći očitati vrijednost potrebne varijable.

```
while (client.connected()) {
```

```
            if (client.available()) {
                  char c = client.read();
                        if (readString.length() < 100 ) {
                        readString += c;
             }
```

Nakon što je varijabla prenesena u *string* može se iskoristiti za postavljanje izlaza. Poziva se funkcija *handle_relay(client);* kako bi se preuzela vrijednost varijable potrebna za upravljanje radom uređaja. To opisuje sljedeći niz naredbi iz funkcije *void handle_relay(EthernetClient cl):*

```
      if (readString.indexOf('r', 5) == 6 && readString.indexOf('2', 11) ==
12){
      relay_stanje = 2;
      cl.println("<h3>Sijalicom upravlja detektor pokreta</h3>");
      }
      else if (readString.indexOf('r', 5) == 6 && readString.indexOf('1', 11)
== 12) {
      relay_stanje = 1;
       }
      else if (readString.indexOf('r', 5) != 6) return;
      else  relay_stanje = 0;
       if (relay_stanje == 1) {
            digitalWrite(relayPin, LOW);
            cl.println("<h3>Sijalica je <b>UPALJENA</b></h3>");
      }
       else if (relay_stanje == 0) {
              digitalWrite(relayPin, HIGH);
             cl.println("<h3>Sijalica je <b>UGASENA</b></h3>");
       }
}
```

Preuzimanje stanja varijable vrši se pomoću metode *.indexOf()*, jer s njom se može pretražiti određenu vrijednost *stringa*, te odrediti od kojeg indexa počninje tražiti. Pregledava se cijela varijabla *readString* koja je zapravo zapisnik znakova koji se slao klijentu, te se traži ime forme *relay=vrijednost(0,1,2)*. Pošto se *get* metodom slala vrijednost forme, *relay* se nalazi na početku varijable *readString*. Ako je slovo 'r' na 6. mjestu i njegova vrijednost '2' na 12. mjestu, varijabla *relay_stanje* postavlja se na vrijednost 2. Na isti način traže se i ostale vrijednosti, čija stanja su prikazana u tablici:

Tablica 3.1

Vrijednost	Stanje
0	Ugašeno
1	Upaljeno
2	Upravljano senzorom

Za razliku od stanja 0 (ugašeno) i 1(upaljeno) koji su postavljeni u funkciji *handle_relay();* stanje 2(upravljano senzorom) moralo se postaviti unutar *loop();* petlje iz razloga što to stanje zahtjeva komunikaciju sa senzorom pokreta.

```
if (relay_stanje == 2) {
            int pirValue;
            pirValue = digitalRead(pirPin);
            digitalWrite(relayPin, !pirValue);
      }
```

4. ZAKLJUČAK

U ovom radu napravljeno je programsko i sklopovsko rješenje za model digitalne sobe uz pomoć Arduino platforme koja se može upravljati putem web preglednika. Za izradu rada koristila se Arduino UNO maketa, Ethernet maketa, HC-SR501 senzor pokreta i 1-kanalni relej. Prilikom izrade programa problem je nastao prilikom čitanja podataka s web servera koji je otklonjen predavanjem ključne varijable u URL vezu. Prilikom obrade podataka korištena je URL veza kako bi se došlo do vrijednosti tražene varijable za daljnje upravljanje. Potrebno je napomenuti kako prilikom rada s opremom postoji opasnost od dolaska u dodir s opasnim naponom gradske mreže, te je potrebno pristupati s oprezom.

LITERATURA

1. https://arduino-info.wikispaces.com/ArduinoPower
   primjer spajanja Arduina s relejima (kolovoz, 2016.)
2. http://www.instructables.com/id/Ethernet-Switching-with-Arduino/?ALLSTEPS
   primjer rada Arduino i Ethernet maketa (rujan, 2016.)
3. http://www.w3schools.com/html/tryit.asp?filename=tryhtml_radio
   izgled i kod HTML buttona (kolovoz, 2016.)
4. http://www.w3schools.com/jsref/jsref_indexof.asp
   općenito o metodi *indexOf()* (kolovoz, 2016.)
5. https://www.arduino.cc/en/Reference/EthernetClient
   osnove o radu s Ethernet klijentu (rujan, 2016.)
6. https://www.arduino.cc/en/Main/ArduinoEthernetShield
   osnove o Ethernet maketi (rujan, 2016.)
7. https://www.arduino.cc/en/Main/ArduinoBoardUno
   osnove o Arduino Uno maketi (kolovoz, 2016.)
8. http://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/
   osnove o senzoru pokreta (rujan, 2016.)

SAŽETAK

Naslov: Digitalna soba

U završnom radu obrađuje se izrada sklopovskog i programskog dijela za model digitalne sobe kojom se udaljeno upravlja pomoću web stranice. Izrada se temelji na Arduino razvojnom sustavu koji služi kao upravljačka jedinica, dok nam Ethernet modul omogućuje pristup maketi, a samim time i upravljanje istom.

Ključne riječi: Arduino, Ethernet, PIR, relej

ABSTRACT

Title: Digital room

This paper elaborates on development of software and hardware of digital room model remotely operated from a web page. The design is based on Arduino system which serves as a control unit, while Ethernet module allows us to access the model and control the entire system.

Key words: Arduino, Ethernet, PIR, relay

ŽIVOTOPIS

Rođen 9. kolovoza 1994. godine u Osijeku od oca Željka i majke Janje. Živi s obitelji u Tenju, gdje pohađa i uspješno završava „OŠ Tenja“. Srednju školu upisuje III. Gimnaziju Osijek koju završava 2012. godine. Iste godine upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija, preddiplomski studij računarstva.

Ivan Kovačić

_____

PRILOG A: Program za Arduino Uno razvojnu maketu

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
byte ip[] = {  192, 168, 1, 102  };
byte subnet[] = {  255, 255, 255, 0 };
byte gateway[] = {  192, 168, 1, 1  };

EthernetServer server(80);

int relay_stanje = 0;
String readString;
int pirPin = 6;
int relayPin = 3;

void handle_relay(EthernetClient cl) {
      cl.println("<input type=\"radio\" name=\"relay\" value=\"1\">On<input
type=\"radio\" name=\"relay\" value=\"0\">Off<input type=\"radio\"
name=\"relay\" value=\"2\">PIR</form>");
      if (readString.indexOf('r', 5) == 6 && readString.indexOf('2', 11) ==
12){
      relay_stanje = 2;
      cl.println("<h3>Sijalicom upravlja detektor pokreta</h3>");
      }
      else if (readString.indexOf('r', 5) == 6 && readString.indexOf('1', 11)
== 12) {
      relay_stanje = 1;
       }
      else if (readString.indexOf('r', 5) != 6) return;
      else  relay_stanje = 0;
       if (relay_stanje == 1) {
            digitalWrite(relayPin, LOW);
            cl.println("<h3>Sijalica je <b>UPALJENA</b></h3>");
      }
       else if (relay_stanje == 0) {
              digitalWrite(relayPin, HIGH);
             cl.println("<h3>Sijalica je <b>UGASENA</b></h3>");
       }
}

void setup() {
       Ethernet.begin(mac, ip, gateway, subnet
       server.begin();
       pinMode(relayPin, OUTPUT);
       digitalWrite(relayPin, HIGH);
       pinMode(pirPin, INPUT);
}
void loop() {
      EthernetClient client = server.available();
      if (relay_stanje == 2) {
            int pirValue;
            pirValue = digitalRead(pirPin);
            digitalWrite(relayPin, !pirValue);
      }
      if (client) {
            boolean currentLineBlank = true;
            while (client.connected()) {
            if (client.available()) {
```

```
                char c = client.read();
                       if (readString.length() < 100 ) {
                      readString += c;
            }
           if (c == '\n' && currentLineBlank) {
                     client.println("HTTP/1.1 200 OK");
                client.println("Content-Type: text/html");
                client.println("Connection: keep-alive");
                client.println("");
                client.println("<!DOCTYPE html>");
                client.println("<html>");
          client.println("<head>");
          client.println("<title>Internet prekidac</title>");
          client.println("</head>");
          client.println("<body>");
          client.println("<h2>Svjetlo: </h2>");
          client.println("<form method=\"get\">");
          client.println("<input type=\"submit\" value=\"Submit\">");
          handle_relay(client);
          client.println("</body>");
          client.println("</html>");
          break;
        }
        if (c == '\n') {
           currentLineBlank = true;
        }
        else if (c != '\r') {
               currentLineBlank = false;
        }
      }
    }
    delay(10);
    client.stop();
    readString = "";
  }
}
```

PRILOG B: Program za internet stranicu

```
<html>
    <head>
        <title>Internet prekidac</title>
    </head>
    <body>
        <h2>Svjetlo: </h2>
        <form method="get">
            <input type="radio" name="relay" value="1">On
            <input type="radio" name="relay" value="0">Off
            <input type="radio" name="relay" value="2">PIR
            <input type="submit" value="Submit">
        </form>
    </body>
</html>
```