



ŠOLSKI CENTER CELJE

Srednja šola za elektrotehniko, kemijo, in računalništvo

NADZOR ELEKTRIČNIH PORABNIKOV

RAZISKOVALNA NALOGA

MENTOR:
Gregor Kramer univ. dipl. inž. el.

Avtor:
Nejc KOVAČIČ, E-4.a

Celje, 2016

1. KAZALO

1.	KAZALO	2
1.1	KAZALO SLIK.....	3
2.	POVZETEK	4
2.1	KLJUČNE BESEDE.....	4
3.	UVOD	5
3.1	HIPOTEZE	5
4.	OPIS RAZISKOVALNE METODE.....	6
5.	OSREDNJI DEL NALOGE.....	6
5.1	PREDSTAVITEV REZULTATOV RAZISKOVANJA	6
5.2	ARDUINO UNO.....	7
5.3	ARDUINO ETHERNET SHIELD	8
5.4	OSEM KANALNI RELE MODUL	9
5.5	IZDELAVA VEZJA.....	9
6.	RAZPRAVA	19
7.	ZAKLJUČEK	20
8.	ZAHVALA.....	21
9.	VIRI IN LITERATURA	22
	IZJAVA	23

1.1 KAZALO SLIK

Slika 1: Idejna shema povezave med komponentami	6
Slika 2: Arduino UNO	7
Slika 3: Arduino UNO izhodi	7
Slika 4: ATmega328P čip	8
Slika 5: Ethernet Shield modul	8
Slika 6: 8 kanalni relejski modul	9
Slika 7: Združitev razvojne ploščice Arduino in modulom Ethernet Shield	10
Slika 8: Idejna skica priključitve relejskega modula na Ethernet Shield	10
Slika 9: Praktična priključitev relejskega modula na Ethernet Shield	11
Slika 10: Idejna shema priključitve porabnika na rele	11
Slika 11: Praktična priključitev relejev in porabnikov (led diod)	12
Slika 12: Knjižnice za prvi program	12
Slika 13: Del kode za spletno stran prvega programa	13
Slika 14: Spletna stran za prvi program	13
Slika 15: Določanje ip-jev v programu	13
Slika 16: Del prvega programa	14
Slika 17: Celotno vezje za prvi program	14
Slika 18: Shema vezave za tipko	15
Slika 19: Shema vezave led diode in upora	15
Slika 20: Praktični del vezave tipke in led diode	16
Slika 21: Knjižnice za drugi program	16
Slika 22: Del kode za spletno stran	17
Slika 23: Spletna stran	17
Slika 24: Nastavitev pinov	17
Slika 25: Del drugega programa	18
Slika 26: Celotna vezava za drugi program	18

2. POVZETEK

V raziskovalni nalogi sem raziskoval krmiljenje oz. nadzor oddaljenih naprav preko spletne strani. Za to nalogo sem se odločil ker me zanima komuniciranje z računalnikom in napravami, ter komuniciranje preko spleta, idejo pa sem dobil na spletni strani. Cilj te naloge, je bil narediti napravo, ki omogoča vklopjanje oz. izklopjanje različnih porabnikov preko spletne strani, ali pa z fizičnim vklopom oz. izklopom. Za doseg tega cilja sem uporabil razvojno ploščo, ter modul, ki je povezal razvojno ploščo in splet, preko katerega sem potem lahko vklopljal in izklopljal porabnike, v mojem primeru LED diode.

Z rezultati raziskave sem bil zelo zadovoljen, saj sem po velikem napenjanju možganov, vse hipoteze in trditve, ki sem jih postavil na začetku, tudi potrdil.

2.1 KLJUČNE BESEDE

ATmega328P – mikroprocesor,

Ethernet Shield – modul

Arduino UNO – razvojna ploščica

Relejski modul

SD kartica

3. UVOD

Na spletu sem najprej poiskal podobne primere nalog, kot je moja, za lažjo orientacijo. Potem sem primerjal svojo željo, kako hočem, da bo delovala moja naloga in že prej narejene podobne naloge, ter dobil neke smernice, kako je potrebno izdelati nalogo in z katerimi problemi se lahko srečam pri izdelavi naloge.

Moja želja je bila izdelati napravo, preko katere, bi lahko krmilil različne naprave, ki jih imam doma (klima, ogrevalni sistemi, luči, garažna vrata..) brez da bi bilo potrebno stopiti do naprave in jo fizično zagnati oz. ustaviti. Da pa bi bilo to varno in dostopno po celi stanovanjski hiši, sem naredil tako, da sem domače omrežje povezal z modulom za internet, kateri je bil že prej povezan z razvojno ploščo. Na razvojno ploščo pa sem napisal program, za vklop in izklop porabnikov preko spleta, ter drugi program ki omogoča vklapljanje in izklapljanje porabnikov tudi ročno ali preko spleta hkrati. Ker je spletna stran, ki omogoča nadzor porabnikov, na domačem omrežju, lahko do spletne strani dostopa samo porabnik, ki ima dostop do tega omrežja.

Za nalogo sem potreboval modul (Ethernet shield), ki povezuje omrežje in razvojno ploščo (Arduino UNO) z ustreznim mikrokrmilnikom. Krmilje pa bo izvedenem v razvojnem okolju Arduino. Ker v slovenskih trgovinah ni bilo moč kupiti ustreznega krmilnika in modula, sem se za te stvari pozanimal na spletu, kjer sem jih tudi kupil. Krmilje naprav pa sem zaradi lažje predstavitve ponazoril z LED diodami.

3.1 HIPOTEZE

- Sistem omogoča daljinsko upravljanje 8 porabnikov preko spletne strani ali lokalno preko fizičnih stikal
- Cena naprave je maksimalno 100€

4. OPIS RAZISKOVALNE METODE

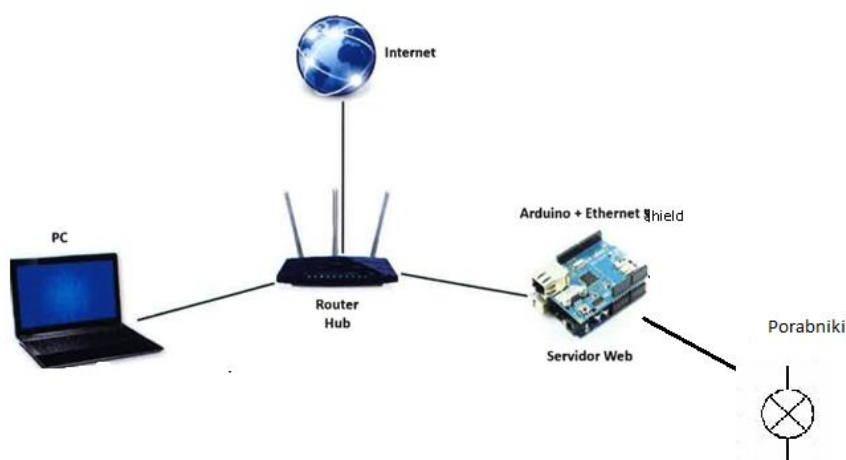
Naloga, ki sem si jo izbral, je bila zahtevna predvsem z programskega vidika, saj sem moral povezati krmilnik z omrežjem. Nalogo sem začel tako, da sem na spletu najprej poiskal krmilnik, s katerim sem se že srečal. Potem pa sem glede na krmilnik, ki sem si ga izbral, iskal internetni modul, ki ga krmilnik podpira ter, da je vse skupaj cenovno ugodno. Našel sem Arduinotov Ethernet Shield, modul za internet ter Arduino UNO krmilnik, zraven pa sem še kupil relejski modul, za lažje preklapljanje in programiranje. V programskem okolju Arduino, sem napisal dva programa, prvi program omogoča nadzor porabnikov sam preko spleta in krmilimo lahko 8 porabnikov preko relejskega modula. V fazi pripravljanja sem za svojo nalogo izbral 8 led diod, ki predstavljajo porabnike. Drugi program pa sem napisal tako, da lahko kontroliramo dva porabnika, in sicer ali preko spleta, ali pa kot fizični vklop in izklop porabnikov z tipkami.

5. OSREDNJI DEL NALOGE

5.1 PREDSTAVITEV REZULTATOV RAZISKOVANJA

Pri tej nalogi, je bil bistveni del vzpostaviti povezavo med računalnikom, krmilnikom, ki je bil povezan z modulom za internet, ter omrežjem.

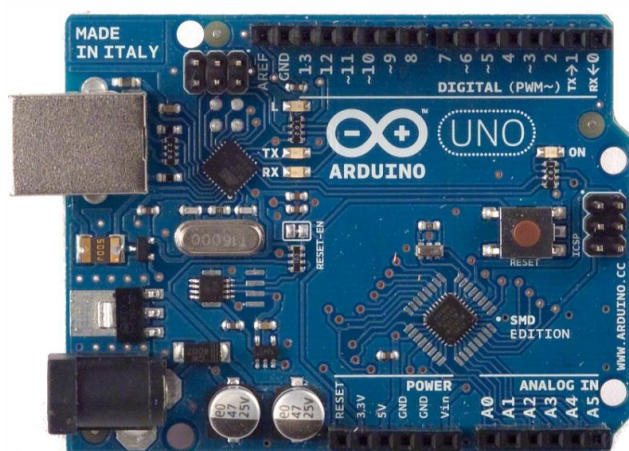
Največji problem je bil programski del, kako povezati omrežje z krmilnim delom.



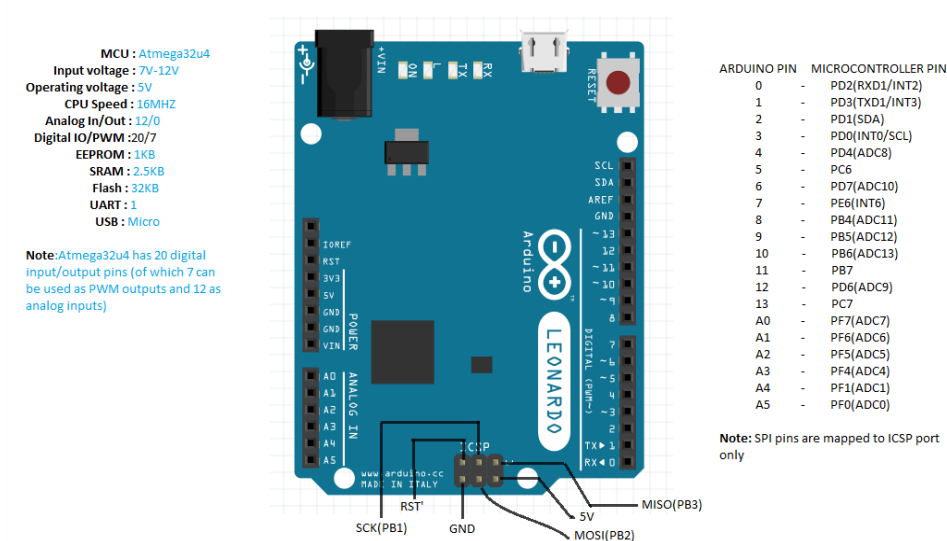
Slika 1: Idejna shema povezave med komponentami

5.2 ARDUINO UNO

Razvojna plošča Arduino Uno temelji na Atmelovem 8-bitnem mikrokontrolerju ATmega 328P. Njene glavne značilnosti: Vsebuje 14 digitalnih vhodno / izhodnih priključkov, od katerih jih lahko 6 uporabimo za PWM signale in 6 kot analognih vhodov. Delovna napetost znaša 5 V, posamezne elemente pa lahko napetostno napajamo, kar iz razvojne plošče. Za to imamo na voljo napetost 5 V in 3,3 V. Paziti moramo, da posameznih izhodov ne preobremenimo. Posamezni izhod lahko obremenimo z največ 40 mA pri napetosti 5 V in z največ 50 mA pri napetosti 3,3 V. Takt diktira 16 MHz oscilator. Vsebuje 32 KB flash programskega pomnilnika, za vpisovanje programa v mikrokontroler. Ta se ne izbriše, če razvojni plošči izklopimo napajanje. Poleg flasha vsebuje še 2 KB RAM in 1 KB EEPROM pomnilnika.



Slika 2: Arduino UNO



Slika 3: Arduino UNO izhodi

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Slika 4: ATmega328P čip

5.3 ARDUINO ETHERNET SHIELD

Arduino Ethernet Shield omogoča Arduino krmilniku, da se poveže z internetom. Temelji na ethernet čipu WIZnet W5100, ki zagotavlja omrežje (ip). Za pisanje programov z Ethernet Shieldom moramo obvezno uporabiti Ethernet knjižnice. Na ploščici je tudi priključek za SD kartico, ki se lahko uporablja za shranjevanje datotek (spletna stran), ki jih potem Arduino bere iz SD kartice preko knjižnice za SD.



Slika 5: Ethernet Shield modul

5.4 OSEM KANALNI RELE MODUL

8 kanalni 5V relejski modul - interface board, ki lahko krmili visoko tokovne porabnike. Krmilite ga direktno z mikrokontrolerja, kot Arduino, 8051, AVR, PIC, DSP, ARM, MSP430, TTL...

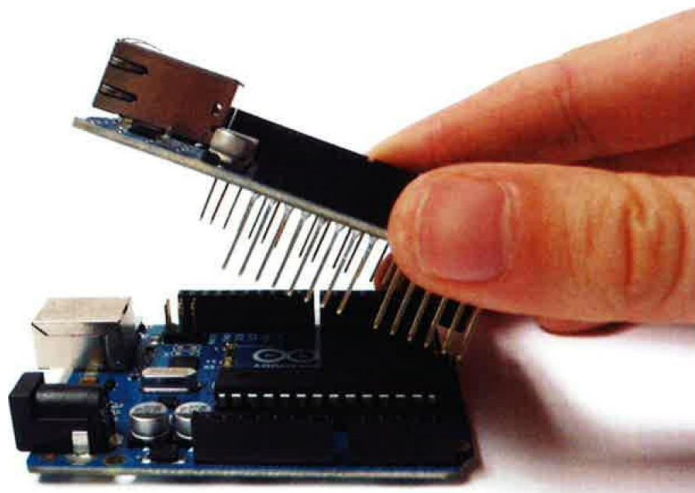
Vsak rele potrebuje 15-20mA krmilnega toka, rele pa ima mirovni in delovni kontakt. Modul je opremljen z releji s kontaktom AC 250V 10A in DC 30V 10A, ima pa tudi LED indikator za prikaz stanja releja.



Slika 6: 8 kanalni relejski modul

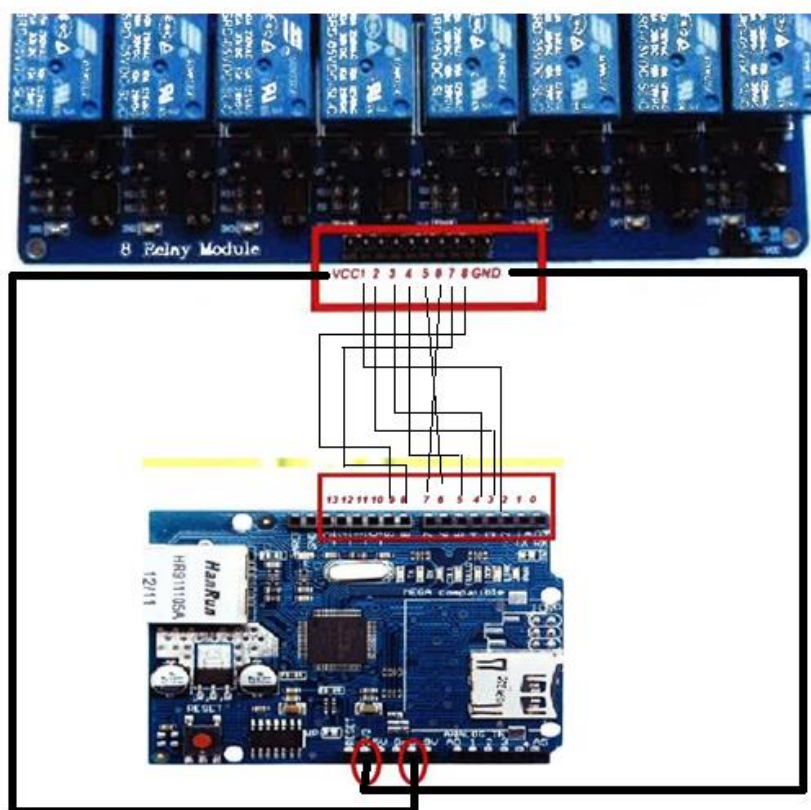
5.5 IZDELAVA VEZJA

Najprej moramo združiti ploščico Arduino UNO in Ethernet Shield, ker sta tovarniško tako narejena, preprosto Ethernet Shield vstavimo v Arduino UNO, kot kaže spodnja slika.

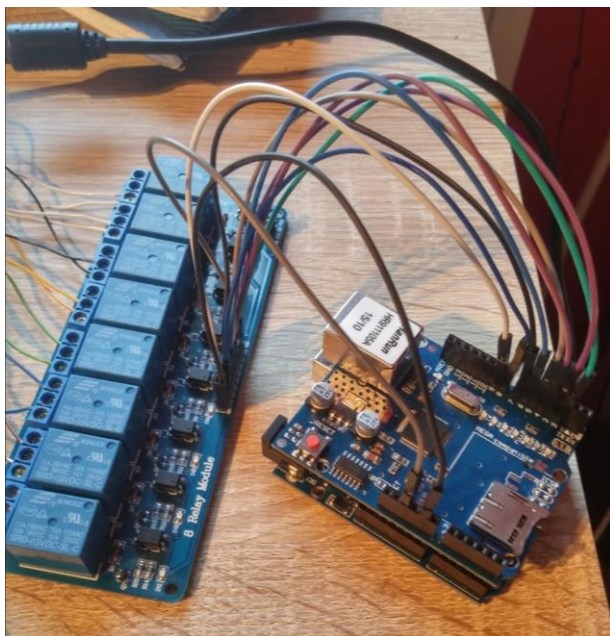


Slika 7: Združitev razvojne ploščice Arduino in modulom Ethernet Shield

Nato pa za program, ki vsebuje osem kanalni rele, releje povežemo z Ethernet Shieldom. Pine od 2 do 9 na Ethernet Shieldu, povežemo z relejskim modulom, na vhode od 1 do 8. Na relejski modul pa moramo še pripeljati 5V (VCC) in maso (GND).



Slika 8: Idejna skica priključitve relejskega modula na Ethernet Shield

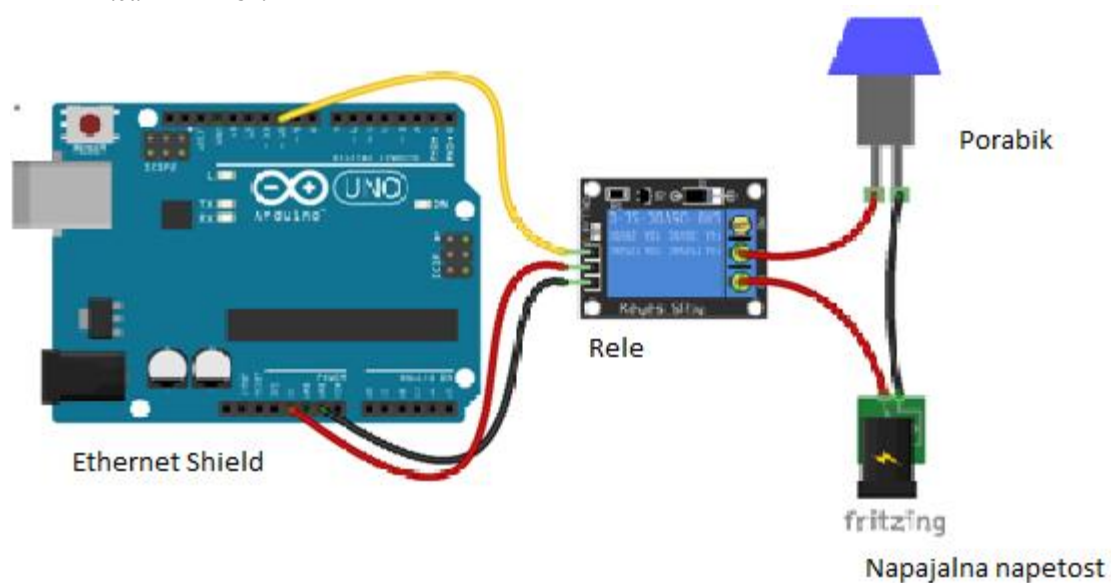


Slika 9: Praktična priključitev relejskega modula na Ethernet Shield

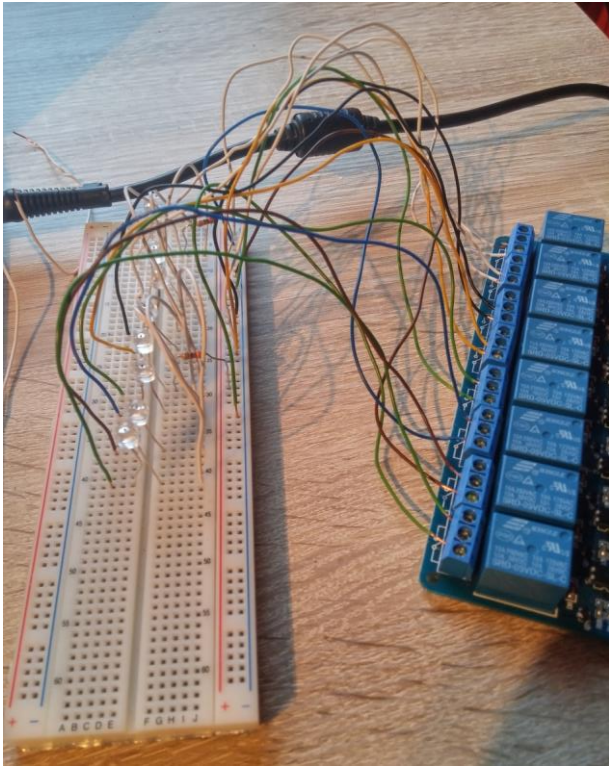
Ko sem relejski modul in Ethernet Shield povezal, sem na izhode relejev vezal 8 led diod z preduporom, ki smo ga izračunali po spodnji formuli.

Vrednost uporov R , za omejitev izhodnega toka in zaščito LED diode, smo izračunali:

$$R = \frac{U_{pin} - U_{led}}{I_{led}} = \frac{5V - 2V}{15mA} = 333\Omega \quad R = 330\Omega$$



Slika 10: Idejna shema priključitve porabnika na rele



Slika 11: Praktična priključitev relejev in porabnikov (led diod)

Naprava potrebuje za svoje delovanje ukaze, ki bodo povezali med sabo vse elemente Arduino UNO, Ethernet Shield in relejski modul. Da bomo to dosegli, moramo v program vstaviti knjižnice.

```
#include <Ethernet.h>

#include <SPI.h>
```

Slika 12: Knjižnice za prvi program

Nato je potrebno v programu napisati spletno stran, za kontrolo osmih porabnikov. S tem ko v program vpišemo kode za spletno stran, določimo kakšen bo imela zgled spletna stran in kaj vse bo vsebovala. Naslov spletne strani pa določim sam, že na začetku, ko vpišem ip, ta ip je tudi naslov spletne strani. To pomeni, da vedno ko bom želel iti na spletno stran in krmiliti porabnike, bom v brskalnik vpisal ta ip.

```

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println();

client.println("<html><body><form method=get>");

client.println("<p>Kontrola porabnikov</p>");

for(int i=0;i < numofreles ;i++){

Rele = String("porabnik") + i;

if((inString.indexOf(Rele+"=on")>0) || inString.indexOf("all=off")>0){

digitalWrite(rele[i], HIGH);

value[i] = "off";
color[i] = "red";

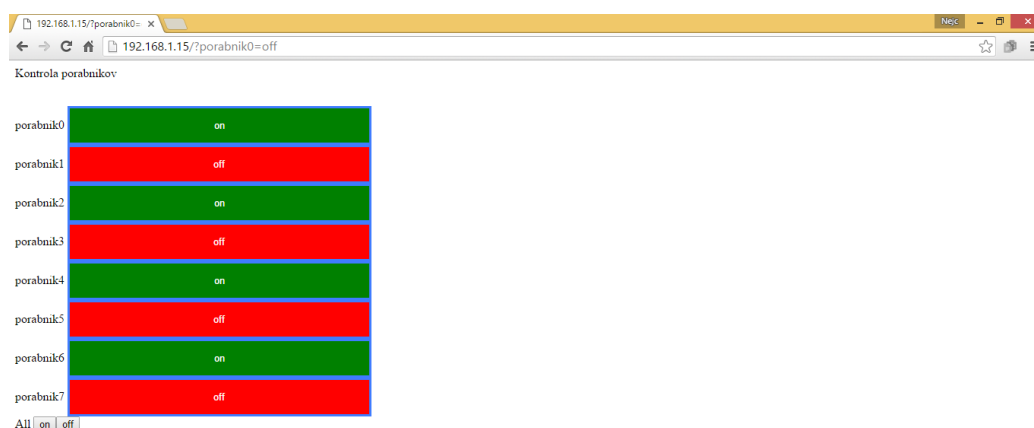
}else if(inString.indexOf(Rele+"=off")>0 || inString.indexOf("all=on")>0 ){

//Serial.println(Rele+"green");

digitalWrite(rele[i], LOW);

```

Slika 13: Del kode za spletno stran prvega programa



Slika 14: Spletna stran za prvi program

Nato pa moramo še v program vpisati ustrezne ip naslove, za povezavo Arduinota z domačim omrežjem, ter objavljanje spletne strani na omrežju. Ip getaway in subnet imamo že vnaprej določena, najdemo jih v programu »cmd«, z ukazom »ipconfig« in jih samo prepíšemo.

```

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

byte ip[] = { 192, 168, 1, 15 };

byte gateway[] = { 192, 168, 1, 1 };

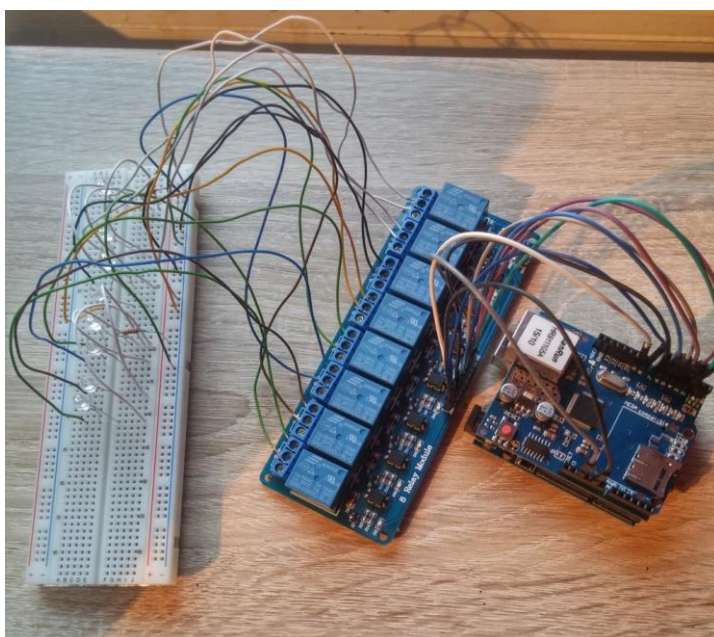
byte subnet[] = { 255, 255, 255, 0 };

```

Slika 15: Določanje ip-jev v programu


```
101$  
String inString = String(35);  
  
String Rele;  
  
int rele[] = {2, 3, 4, 5, 6, 7, 8, 9};  
  
int numofreles = 8; //numofreles  
  
String value[] = {"on","on","on","on","on","on","on","on"};  
String color[] = {"green","green","green","green","green","green","green","green"};  
  
EthernetServer server(80);  
  
String data;  
void setup()  
{  
  
  Serial.begin(9600);  
  Ethernet.begin(mac,ip,gateway,subnet);  
  server.begin();  
  for (int j = 0; j < (numofreles); j++){  
    |  
    pinMode(rele[j], OUTPUT);  
  }  
}  
  
Nalaganje končano  
  
Skica uporablja 15.610 bajtov, kar je (48%) prostora namenjenega programu. Maksimum je 32.256 bajtov.  
Globalna spremenljivka uporablja 986 bajtov, kar je (48%) dinamičnega spomina in pušča 1.062 bajtov za lokalne spremenljivke. Maksimum je 2.048 bajtov.
```

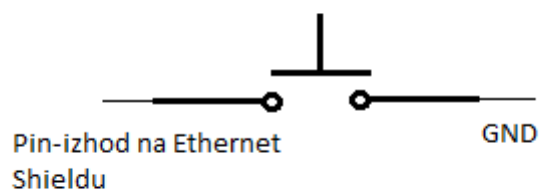
Slika 16:Del prvega programa



Slika 17:Celotno vezje za prvi program

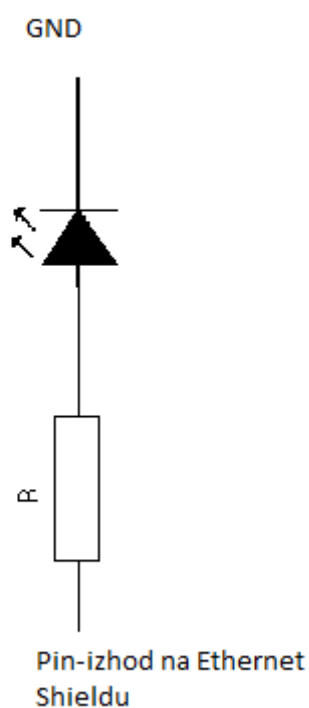
Za izdelavo drugega programa pa nisem uporabil relejskega modula, ampak sem najprej združil Arduino UNO in Ethernet Shield, kot že pri prejšnjem programu. Nato sem na izhoda 2 in 3 vezal tipki, kateri bojo omogočali vklop in izklop porabnikov fizično, preko teh dveh tipk. Na izhoda 6 in 7 pa sem vezal dve led diode, ki predstavljata porabnika, z pred uporom, ki sem ga izračunal po formuli:

$$R = \frac{U_{pin} - U_{led}}{I_{led}} = \frac{5V - 2V}{15mA} = 333\Omega \quad R = 330\Omega$$

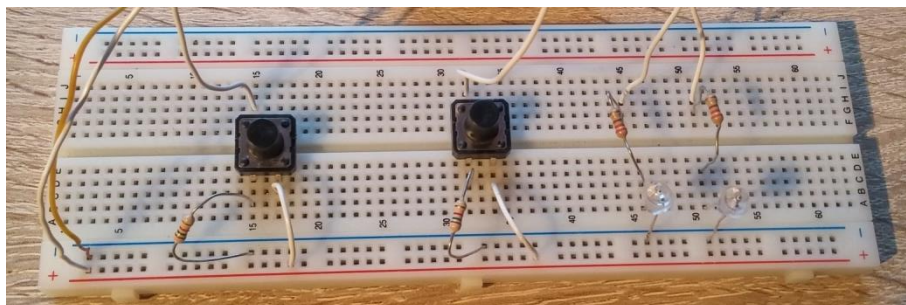


Slika 18: Shema vezave za tipko

Led diodo pa sem vezal:



Slika 19: Shema vezave led diode in upora



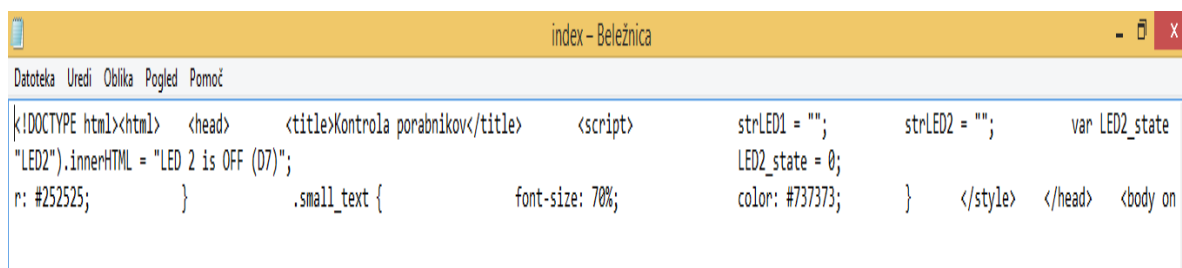
Slika 20:Praktični del vezave tipke in led diode

Za delovanje te naloge, pa smo morali v program prav tako vstaviti določene knjižnice, ki omogoča povezovanje vseh potrebnih elementov.

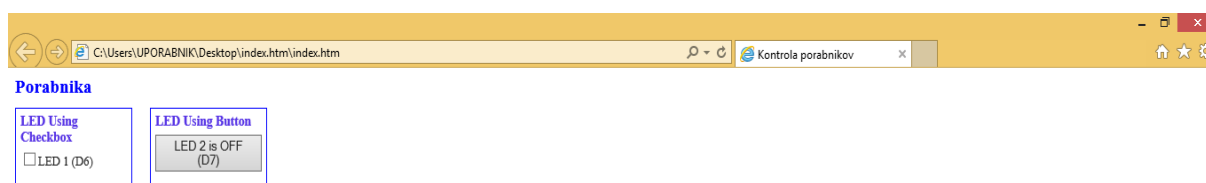
```
-----  
  
#include <SPI.h>  
#include <Ethernet.h>  
#include <SD.h>  
|
```

Slika 21:Knjižnice za drugi program

Nato je bilo potrebno narediti spletno stran, ki pa je nisem napisal v program, ampak sem jo posebej naredil. Sam sem naredil spletno stran z html kodo, ki sem jo pisal v beležnico z ustreznimi ukazi, ki sem se jih naučil na spletu. To spletno stran je bilo potrebno shraniti pod »index.htm«, kajti samo pod to oznako Arduino bere spletno stran. Nato sem naložil spletno stran na SD kartico, sam sem uporabil 2GB SD kartico, nato sem vstavil SD kartico v Ethernet Shield. Da bomo spletno stran našli na omrežju, moramo prav tako sami določiti ip, ki ga vpišem na začetku programa, da pa želim priti na spletno stran, moram pa v brskalnik vpisati ip, ki smo ga določili na začetku.



Slika 22: Del kode za spletno stran



Slika 23: Spletna stran

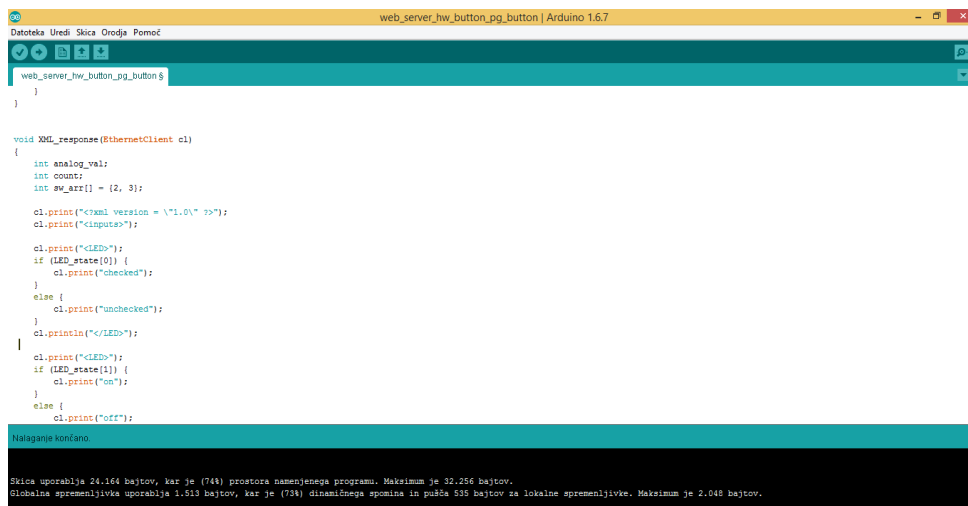
V programu je bilo nato potrebno nastaviti pina 2 in 3 kot vhoda, na katera sem vezal dve stikali za vklop in izklop led diode, pina 6 in 7 pa sem nastavljal kot izhoda, ter sem nanju vezal 2 led diode, ki jih lahko vklopim in izklopim preko spleta ali pa fizično preko dveh tipk.

```

}
Serial.println("SUCCESS - Found index.htm file.");
// switches
pinMode(2, INPUT);
pinMode(3, INPUT);
// LEDs
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);

```

Slika 24: Nastavitev pinov



```
web_server_hw_button_pg_button 5
}

void XM_response(EthernetClient cl)
{
  int analog_val;
  int count;
  int sv_arr[] = {2, 3};

  cl.print("<?xml version = \"1.0\" ?>");
  cl.print("<input?>");

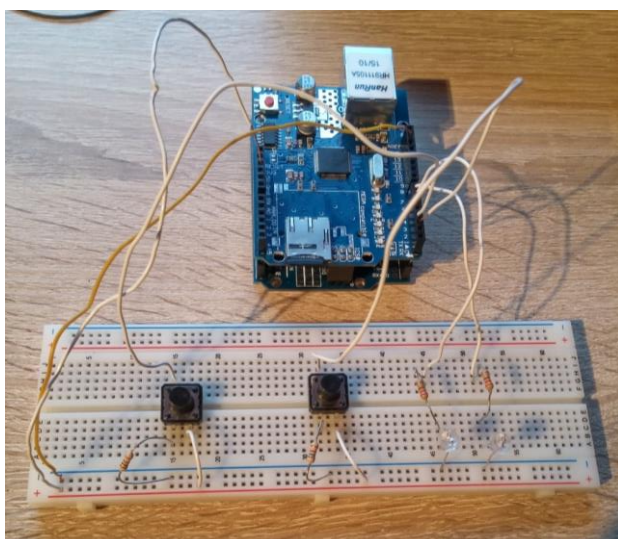
  cl.print("<LED?>");
  if (LED_state[0]) {
    cl.print("checked");
  }
  else {
    cl.print("unchecked");
  }
  cl.println("</LED?>");

  cl.print("<LED?>");
  if (LED_state[1]) {
    cl.print("on");
  }
  else {
    cl.print("off");
  }
}

// Načrtovanje končano.

// Skica uporablja 24.164 bajtov, kar je (74%) prostora namenjenga programu. Maksimum je 32.256 bajtov.
// Globalna spremenljivka uporablja 1.913 bajtov, kar je (73%) dinamičnega spomina in pušča 939 bajtov za lokalne spremenljivke. Maksimum je 2.048 bajtov.
```

Slika 25:Del drugega programa



Slika 26:Celotna vezava za drugi program

6. RAZPRAVA

Raziskovalno nalogo sem uspešno končal, saj sem potrdil vse hipoteze, ki sem si jih zadal na začetku raziskovalne naloge.

Porabnike lahko sedaj po končani nalogi nadzorujem in upravljam preko spleta, je varno za uporabo, saj deluje preko domačega omrežja, do katerega pa imamo dostop samo doma, bodisi preko wifija ali napravo v katero je priključen utp kabel, iz domačega routerja. Naloga omogoča nadzorovanje osmih izhodov, ki so vezani na rele in iz releja na porabnike, nadzorujemo jih pa preko spleta. Omogoča pa tudi nadzor dveh porabnikov, ki jih lahko nadzorujemo bodisi preko spleta, ali pa fizično preko dveh tipk. Naprava je cenovno ugodna, saj bi odšteli za ves material približno maksimalno 50€.

7. ZAKLJUČEK

Raziskovala naloga je končana in lahko rečem, da sem zadovoljen, da mi jo je uspelo narediti v celoti. Glede na to, da je na trgu kup podobnih stvari, lahko rečem, da sem zadovoljen z končnim izdelkom glede na to, da sem naredil vse sam za dokaj malo denarja.

Ob tej nalogi sem videl koliko znanja je potrebno za takšen projekt. Videl pa sem še, da se da narediti, z malo znanja o programiranju in vedenju o mikro krmilnikih, zelo veliko in to za veliko manjši znesek.

8. ZAHVALA

Zahvaljujem se profesorju Gregorju Kramerju, uni. dipl. inž. el. za pomoč pri razvijanju ter prilagajanju vezja in programa, pa tudi za podporo in njegov trud skozi celoten projekt.

Prav tako se zahvaljujem vsem profesorjem, ki so mi pomagali s koristnimi nasveti ter mojemu prijatelju, ki mi je pomagal uresničiti, da je naloga dokončana.

9. VIRI IN LITERATURA

ArduinoUNO:

<https://www.arduino.cc/en/Main/ArduinoBoardUno>

Ethernet Shield:

<https://www.arduino.cc/en/Main/ArduinoEthernetShield>

Relejski modul:

https://www.kupindo.com/Elektronika/31009421_ARDUINO-modul-sa-8-releja

ATmega328P:

http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_complete.pdf

IZJAVA

Mentor, Gregor Kramer, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom

Nadzor električnih porabnikov
katere avtor je Nejc Koračič:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisnimi pogoji projekta Mladi za Celje.

Celje, 14.3.2016



Podpis mentorja

Podpis odgovorne osebe

***POJASNILO**

V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je treba podpisano izjavo mentorja(-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja(-ice) fotografskega gradiva, katerega ni avtor(-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

DOVOLJENJE ZA OBJAVO AVTORSKE FOTOGRAFIJE V RAZISKOVALNI NALOGI

Podpisani, Nejc Kovačič, izjavljam, da sem avtor fotografskega gradiva, navedenega v priloženem seznamu, in dovoljujem v skladu z 2. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, da se lahko uporabi pri pripravi raziskovalne naloge pod mentorstvom Gregorja Kramarja z naslovom Nadzor električnih porabnikov, katere avtor je Nejc Kovačič : Dovoljujem tudi, da sme Osrednja knjižnica Celje vključeno fotografsko gradivo v raziskovalno nalogo objaviti na knjižničnih portalih z navedbo avtorstva v skladu s standardi bibliografske obdelave.

Celje, 14.3.2016

Podpis avtorja:
Kovačič