



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradu informacija
Sveučilište u Zagrebu

Sigurna i upravljiva utičnica



- Δ Dio projekta „Pametna kuća“
- Δ Svima zainteresiranima
- Δ Arduino, relej, programiranje
- Δ Povezivanje preko Arduino Ethernet Shielda

Sažetak

Sustav sigurne i upravljive utičnice omogućuje stanarima „pametne kuće“ upravljanje stanjem pametne utičnice na daljinu pomoću web aplikacije. Paljenje i gašenje utičnice na daljinu zapravo nudi mogućnost upravljanja stanjem bilo kojeg trošila priključenog na utičnicu.

Postojeća rješenja su još uvijek nedovoljno popularizirana i relativno skupa u odnosu na realizaciju opisanu u sklopu ove dokumentacije, što je bila glavna motivacija za izradu ovog projekta.

Opisani sustav pametne utičnice sastoji se od sklopovske i programske realizacije. Sklopovska realizacija podrazumijeva ugradnju tiskane pločice s relejem unutar kućišta produžnog kabela koji zatim preko Arduina i Ethernet Shielda komunicira s web aplikacijom. Programska realizacija sustava zapravo podrazumijeva programiranje Arduina kako bi on pravovremeno pametnoj utičnici prosljeđivao zahtjeve korisnika.

Sustav pametne utičnice je primjenjiv u svakom kućanstvu koje ima internetsku vezu te ga je moguće koristiti sa svakog uređaja koji se može povezati na Internet. Korist od sustava mogu imati svi koji uz relativno malo uloženi sredstava žele u svom domu implementirati novu funkcionalnost upravljanja utičnicama i trošilima na daljinu ili u unaprijed definirano vrijeme.

Sadržaj

1. UVOD.....	3
2. OPIS SUSTAVA.....	4
3. SKLOPOVSKA REALIZACIJA SUSTAVA.....	5
3.1. Električna shema.....	5
3.2. Mehaničko rješenje.....	6
3.3. Arduino.....	7
3.4. Arduino Ethernet Shield	8
4. PROGRAMSKA REALIZACIJA SUSTAVA.....	10
4.1. Ručno upravljanje stanjem utičnice	10
4.2. Upravljanje stanjem utičnice pomoću schedulera.....	12
5. REALIZACIJA CIJELOG SUSTAVA.....	14
6. ZAKLJUČAK.....	15
7. LITERATURA.....	16
8. POJMOVNIK	17

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist. a rezultirajuće dielo daie na korištenie pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Temeljna uloga tehnologije i uzrok njenog razvoja je oduvijek bilo ispunjavanje ljudskih potreba i olakšavanje svakodnevnog života, bez obzira radi li se o uštedi vremena, o sigurnosti, praktičnosti ili jednostavno o povećanju udobnosti. Inteligentna ili pametna kuća je izraz koji obuhvaća sva pametna rješenja koja svakodnevni život čine jednostavnijim, a time i produktivnijim. Konačni cilj svakog takvog pametnog rješenja je automatizirati ili u potpunosti eliminirati sve one svakodnevne radnje koje vremenski, prostorno, sigurnosno ili na bilo koji drugi način ograničavaju te tako pozitivno utjecati na način života.

Jedno od takvih pametnih rješenja je i sigurna, upravljiva utičnica. Električne utičnice su sastavni dio svakog kućanstva, a osim što opskrbljuju veliki broj uređaja električnom energijom, njihov potencijal je još uvijek u većem dijelu kućanstava neiskorišten.

Električne utičnice koje su stalno aktivne imaju niz nedostataka. Prije svega, veliki broj uređaja ne zahtjeva neprestano napajanje, posebno zato što tada i u slučaju mirovanja troše određenu količinu električne energije. Isto tako, aktivne utičnice predstavljaju određeni sigurnosti rizik, bez obzira radi li se o maloj djeci ili kućnim ljubimcima. Budući da su električne utičnice većinom standardizirane, one sadrže i određeni potencijal. Naime, često je teško ispuniti želju korisnika da s većinom kućanskih aparata komuniciraju na daljinu te njima upravljaju npr. pomoću mobilnih ili web aplikacija, zbog toga što je standardizacija takvih programa i modula komplicirana, jer se radi o cijelom nizu različitih uređaja i proizvođača. Za razliku od standardizacije različitih uređaja, električne utičnice nude mogućnost razvoja jednog modula koji će omogućiti aktivaciju, odnosno deaktivaciju utičnice, a time i uređaja koji je na nju priključen.

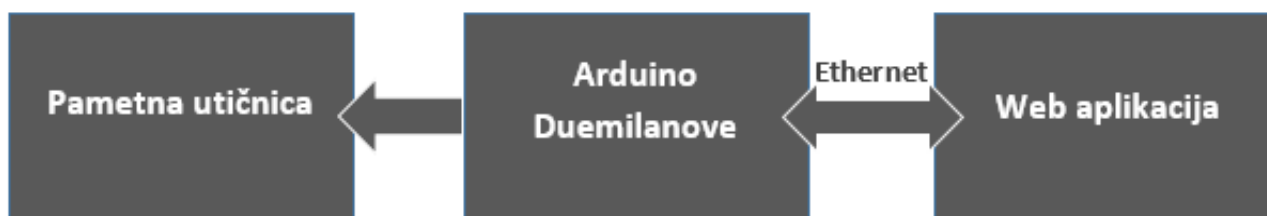
Dodavanjem jednostavnog sklopovlja i programiranjem utičnice, otvara se cijeli niz novih mogućnosti njenog korištenja. Prije svega moguće ju je paliti ili gasiti pomoću bilo kojeg uređaja koji ima pristup Internetu. Malo kvalitetnija aplikacija može omogućiti i paljenje/gašenje utičnice u točno unaprijed definirano vrijeme, kao i periodično paljenje/gašenje u ovisnosti o danu u tjednu ili različitim periodima tijekom dana.

U nastavku su detaljno opisani koraci implementacije gore spomenutih funkcionalnosti pametne utičnice, kao i mogućnosti za daljnji razvoj.

2. Opis sustava

Sigurna i upravljiva utičnica je zapravo naziv cijelog sustava koji na daljinu upravlja utičnicom (tj. produžnim kabelom) i time pali/gasi uređaj spojen na nju. Također, sustav nudi mogućnost paljenja i gašenja utičnice u unaprijed definirano vrijeme (scheduling).

Sustav se sastoji od cjelina prikazanih na Slici 1.



Slika 1. Prikaz sustava sigurne i upravljive utičnice

Kao što je prikazano na slici, pametna utičnica predstavlja sklopovski dio rješenja koji će detaljnije biti opisan u trećem poglavlju. Pametna utičnica je zatim povezana s Arduinoom koji između ostalog upravlja paljenjem i gašenjem utičnice, koristeći pritom upravljački pin. Arduino je također putem Etherneta povezan s Web aplikacijom koja korisniku omogućava jednostavno upravljanje i programiranje pametne utičnice.

3. Sklopovska realizacija sustava

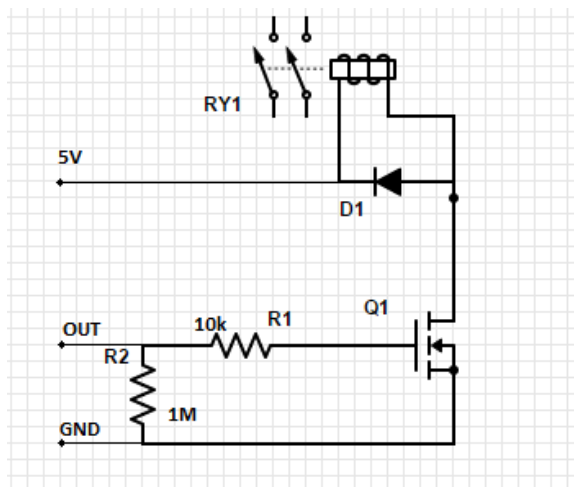
Proces sklopovske realizacije sustava se odnosi na projektiranje električne sheme releja, ugrađivanje dobivene tiskane pločice unutar kućišta produžnog kabela (detaljnije objašnjeno u sklopu poglavlja 3.2.) te povezivanje produžnog kabela s ugrađenim modulom na Arduino i Arduino Ethernet Shield (objašnjeno u poglavljima 3.3. i 3.4.).

3.1. Električna shema

U sklopu projekta odlučili smo, umjesto intervencije na zidnoj utičnici, koristiti produžni kabel, jer je prenosiv, a sastoji se i od kućišta u koje je lako ugraditi dodatne elemente, što je uz zahtjeve ovog projekta bilo neophodno.

Kako bismo uobičajeni produžni kabel pretvorili u sklop kojim je moguće upravljati, bilo je potrebno ugraditi relej, odnosno upravljivu sklopku, unutar same konstrukcije. Uzimajući u obzir potrebne zahtjeve i ograničenja, odabrali smo relej čiji se upravljački pin spaja na 5V (koje Arduino može generirati), a istovremeno podnosi do 220V napona (šifra releja (Chipoteka): 3701223305).

Kako bi odabrani relej optimalno radio, ručno smo ga zalemili na tiskanu pločicu prema shemi na Slici 2.



Slika 2. Električna shema tiskane pločice

Relej je potrebno povezati na način prikazan na Slici 2., a uloga svake od komponenti prikazanih na slici navedena je u nastavku.

- Otpornik ($10\text{k}\Omega$) – predstavlja zaštitu za Arduino u slučaju proboja na desnoj strani električne sheme
- Otpornik ($1\text{M}\Omega$) – u slučaju isključivanja Arduina, osigurava „gašenje“ desne strane električne sheme
- MOSFET – korišten radi jednostavnosti kako struja koju tranzistor daje ne bi ovisila o struji baze koja bi pritom opterećivala izlaz iz Arduina (u slučaju npr. bipolarnog tranzistora). Korišten je MOSFET koji se uključuje na 4V, pri čemu mu je dovoljno dovesti digitalni logički izlaz iz Arduina (0 ili 5V).
- Dioda – predstavlja zaštitu tranzistora od prenapona koji bi nastao u trenutku isključivanja releja (ne dozvoljava da napon na zavojnici releja prijeđe iznos napona napajanja sklopa)

Na slici su vidljivi i pinovi 5V, OUT i GND kojima električni sklop povezujemo s Arduinoom te pinovi releja koje spajamo s postojećim sklopom unutar produžnog kabela.

3.2. Mehaničko rješenje

Nakon što smo ručno zalemili spomenute komponente prema shemi navedenoj na Slici 2., željeli smo nastali sklop što bolje uklopiti unutar postojećeg kućišta produžnog kabela.

Prije svega, bilo je potrebno relej povezati s postojećim sklopom unutar produžnog kabela. Odlučili smo relej spojiti u strujni krug prije klasične zaštitne sklopke koju danas ima većina produžnih kabela, jer na taj način već na temelju svijetljenja lampice unutar sklopke možemo odrediti radi li relej ili ne, odnosno možemo „pročitati“ stanje upravljačkog pina Arduina i utičnice bez potrebe da na produžni kabel priključimo trošilo.

Uz manje intervencije unutar postojećeg kućišta (rezanje plastike i uklanjanje pojedinih dijelova koji osiguravaju stabilnost konstrukcije, zbog čega nije preporučljivo produžni kabel držati na podu zbog opasnosti od gaženja i spoticanja), bilo je moguće uklopiti električni sklop unutar samog produžnog kabela te je rješenje u konačnici praktičnije,

jednostavnije, a i ljepše. Konačni izgled produžnog kabela, odnosno pametne utičnice, moguće je vidjeti na Slici 3.



Slika 3. Produžni kabel s ugrađenim relejem

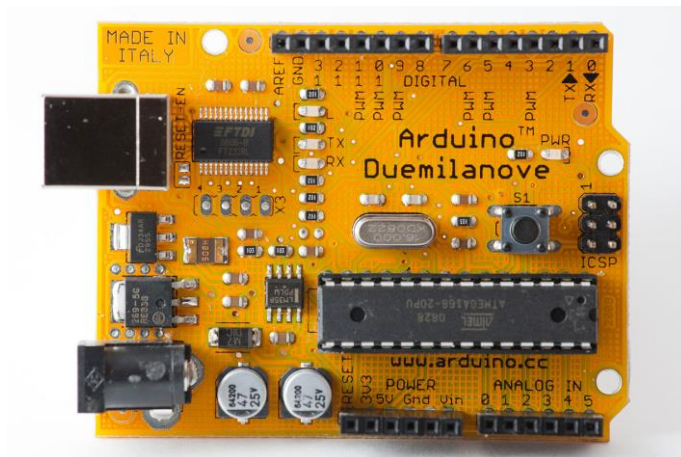
Kao što je vidljivo na Slici 3., dodatni izlazi koje smo dobili modifikacijom produžnog kabela služe za njegovo povezivanje s Arduinoom. Crvenu žicu (Slika 3.) označenu kao 5V priključujemo na pin označen također s 5V. Crna žica predstavlja uzemljenje te ju priključujemo na pin označen s GND. Bijela žica predstavlja upravljački pin čije stanje određuje stanje releja, odnosno stanje utičnice (upaljena/ugašena) te ju priključujemo na pin 8 (definirano programskim kodom).

3.3. Arduino

Arduino je mikrokontroler otvorenog koda, namijenjen početnicima i često korišten kao uvod u procesno programiranje zbog svoje jednostavnosti i pristupačnosti. Arduino platformu moguće je povezati s gotovo svim računalima, a Arduino software moguće je nadograditi i C++ knjižnicama, u slučaju naprednijih korisnika. Glavna uloga Arduina je upravljanje i povezivanje s različitim elektroničkim uređajima u različite svrhe te jednostavan razvoj novih sustava.

U sklopu projekta koristili smo Arduino Duemilanove, što je zapravo tiskana pločica u čijem se središtu nalazi Atmega168 mikrokontroler. Arduino Duemilanove ima 14 digitalnih I/O pinova, 6 analognih ulaza,

16MHz kristalni oscilator, USB vezu, jack za posebno napajanje i tipku za reset (Slika 4.).

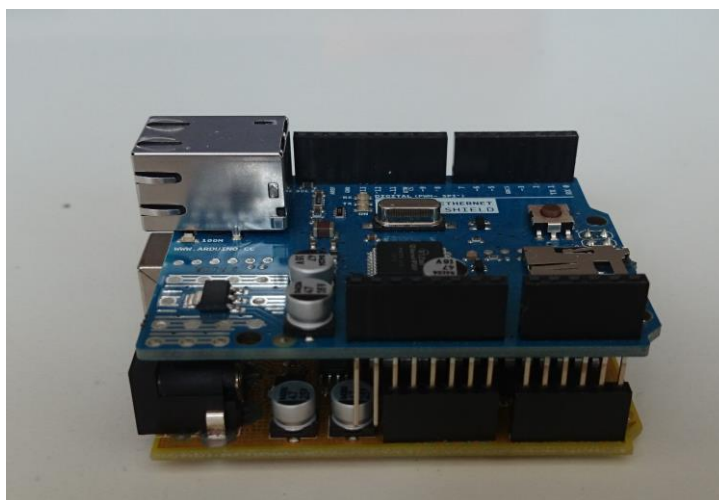


Slika 4. Arduino Duemilanove

Arduino pločica zapravo predstavlja centralni dio sustava koji posreduje između jednostavne Internet aplikacije i pametne utičnice, udovoljavajući na taj način željama korisnika. Na Arduino se osim produžnog kabela, spaja i Ethernet Shield koji je detaljnije opisan u sljedećem poglavlju.

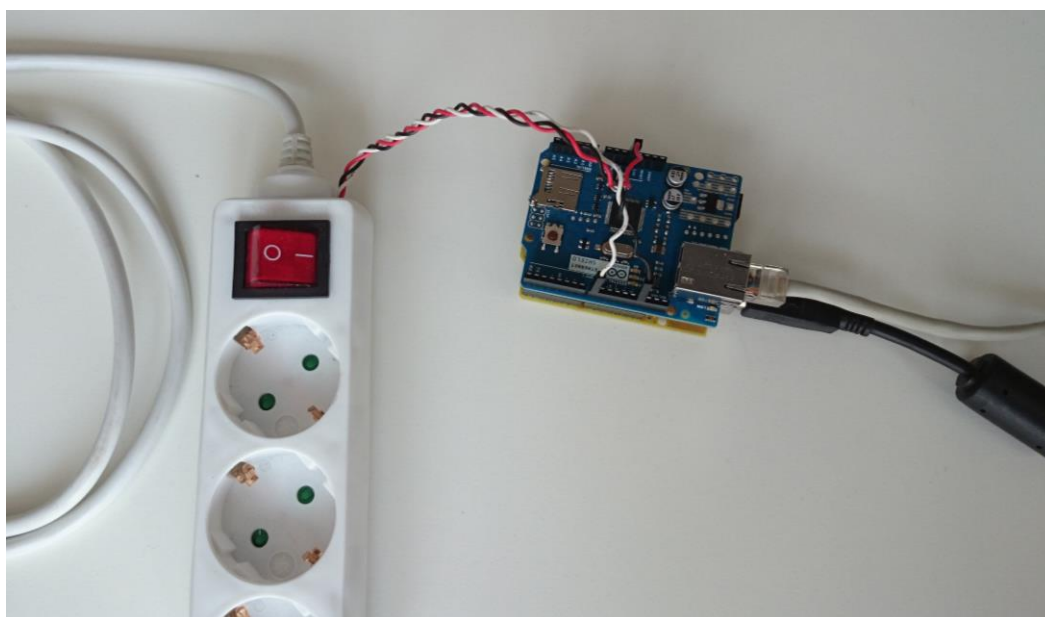
3.4. Arduino Ethernet Shield

Arduino Ethernet Shield je dodatni modul koji Arduino omogućava povezivanje s internetom. Rad Ethernet Shielda temelji se na Wiznet W5100 Ethernet čipu koji omogućava i TCP i UDP protokole.



Slika 5. Arduino Ethernet Shield

Ethernet Shield se na jednostavan način povezuje s Arduinoom, kao što je moguće vidjeti na Slici xy. Ethernet Shield zatim pomoću Ethernet kabela povezujemo na lokalnu mrežu, omogućujući Arduinou povezivanje s internetom. Time je zapravo zaokružena implementacija sklopovskog dijela sustava sigurne i upravljive utičnice. Konačni izgled sustava, zajedno s povezanim podsustavima vidljiv je na Slici 6.



Slika 6. Prikaz sustava sigurne i upravljive utičnice

U sljedećem poglavlju opisana je programska realizacija sustava koja nudi dvije opcije upravljanja stanjem pametne utičnice, ručno ili pomoću schedulera.

4. Programska realizacija sustava

Proces programske realizacije sustava se odnosi prvenstveno na programiranje samog Arduina. U sklopu projekta implementirane su dvije mogućnosti upravljanja sigurnom i upravljivom utičnicom, ručno, koristeći ponuđene opcije u web aplikaciji te pomoću schedulera koji omogućuje paljenje i gašenje utičnice u unaprijed definirano vrijeme.

Programska realizacija sustava temelji se na komunikaciji sustava s web aplikacijom preko koje se vrši upravljanje pametnom utičnicom. Web aplikacija se nalazi na serveru na koji je sustav spojen preko Ethernet modula. Kod koji je korišten prilikom programiranja Arduina naveden je u nastavku, zajedno s objašnjenjima njegovih pojedinih dijelova.

4.1. Ručno upravljanje stanjem utičnice

Programski kod započinje deklaracijom varijabli koje predstavljaju MAC adresu Ethernet Shileda te IP adresu servera na koji se želimo spojiti. Bitno je da IP adresa pripada opsegu mrežnog adaptera na koji je spojen Ethernet modul. Nakon toga se inicijalizira Ethernet uređaja te se definira da se u trenutku poziva spaja na port 80 na kojem server sluša postoje li dolazne poruke.

Također, potrebno je inicijalizirati serijsku vezu tako da sustav može upravljati stanjem releja te dobivati podatke o njegovom trenutnom stanju. Zatim slijedi petlja koja se periodički izvršava i sastoji se od čekanja trenutka spajanja servera i slanja podataka. U petlji se također detektira koja je od dvije opcije u aplikaciji odabrana (uključiti ili isključiti) te se na temelju signala koji se očita, na serijski priključak šalje logička jedinica ili nula.

```
#include <SPI.h>
#include <Ethernet.h>

int relej = 8;

byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0xA8, 0x34 };
byte ip[] = { 161, 53, 64, 35 };
byte gateway[] = { 161, 53, 64, 1 };
byte subnet[] = { 255, 255, 255, 0 };
EthernetServer server(80);
String readString;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(relej, OUTPUT);

  // start the ethernet and server
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin();
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();

        //read char by char HTTP request
        if (readString.length() < 100) {
          //store characters to string
          readString += c;
          //Serial.print(c);
        }

        if (c == '\n') {
          client.println("HTTP/1.1 200 OK"); //new page
          client.println("Content-Type: text/html");
          client.println();
          client.println("<HTML>");
          client.println("<HEAD>");
          client.println("<meta charset='UTF-8'>");
          client.println("<TITLE>Daljinsko upravljanje utičnicom</TITLE>");
          client.println("</HEAD>");
          client.println("<BODY>");
          client.println("<h1>S2U2 - Daljinski upravljiva utičnica</h1>");
          client.println("<h3>SPVP - 2015</h3>");
          client.println("<br />");
          client.println("<a href='\"/?buttonlon\"'>Uključi</a>");
          client.println("<a href='\"/?buttonloff\"'>Isključi</a><br />");
          client.println("<br />");
          client.println("</BODY>");
          client.println("</HTML>");
          client.stop();
          if (readString.indexOf("<?buttonlon") >0){
            digitalWrite(relej, HIGH);
          }
          if (readString.indexOf("<?buttonloff") >0){
            digitalWrite(relej, LOW);
          }
          readString="";
        }
      }
    }
  }
}
```

4.2. Upravljanje stanjem utičnice pomoću schedulera

Programski kod koji omogućuje definiranje vremena paljenja, odnosno gašenja utičnice unaprijed, je nešto drugačiji. Naime, Arduino pločica na sebi ne sadrži real time clock, zbog čega Arduino vrijeme uvijek počinje računati od 1. siječnja 1970. godine (Unix time). Ako želimo da Arduino ima informaciju o stvarnom vremenu (što nam je nužno prilikom korištenja schedulera), možemo ili kupiti dodatni RTC modul za Arduino (DS1302) ili pak možemo odabrati jednostavniju i povoljniju opciju, a to je da Arduino čita vrijeme s interneta (s obzirom da je Ethernet veza već realizirana).

U sklopu projekta odlučili smo podatke o trenutnom vremenu uzimati sa servera i na taj način omogućiti Arduino da uspoređuje unaprijed definirano vrijeme s trenutnim vremenom i na temelju toga upravlja stanjem utičnice.

Zbog duljine koda, u nastavku će biti objašnjeni samo neki njegovi dijelovi, dok će cjelovita verzija koda biti priložena u sklopu dokumentacije.

Niže navedeni odsječak programskog koda definira server s kojeg želimo čitati podatke o trenutnom vremenu te dobiveno vrijeme pomoću varijable `timeZoneOffset` prilagođava našoj vremenskoj zoni. Također, definiramo sinkroniziranje Arduina sa serverom svakih 3600 sekundi, što je potrebno u fazi testiranja, a kasnije je razumno staviti duži vremenski period.

```
/* us.pool.ntp.org NTP server */
IPAddress timeServer(178, 218, 172, 164);
const long timeZoneOffset = 7200L;

/* Syncs to NTP server every 3600 seconds */
unsigned int ntpSyncTime = 3600;
```

Sljedeći dio programskog koda uspostavlja vezu sa serverom te ispisuje poruku o uspješnosti povezivanja. Također, u void `setup()` petlji definiramo i vrijeme u koje želimo upaliti ili ugaziti utičnicu. U nastavku koda Arduino razmjenjuje pakete sa serverom, prilagođava format paketa te u konačnici sprema podatak o stvarnom vremenu u obliku varijable `epoch`, pomoću koje postavlja stvarno vrijeme sustava (`setTime(epoch)`).

```
void setup() {
  Serial.begin(9600);
  int i = 0;
  int DHCP = 0;
  DHCP = Ethernet.begin(mac);
  //Try to get dhcp settings 30 times before giving up
  while( DHCP == 0 && i < 30){
    delay(1000);
    DHCP = Ethernet.begin(mac);
    i++;
  }
  if(!DHCP){
    Serial.println("DHCP FAILED");
    for(;;); //Infinite loop because DHCP Failed
  }
  Serial.println("DHCP Success");

  //Try to get the date and time
  int trys=0;
  while(!getTimeAndDate() && trys<10) {
    trys++;
  }
  Alarm.alarmRepeat(17,45,0,relej_on);
  Alarm.alarmRepeat(18,00,0,relej_off);
}

int getTimeAndDate() {
  int flag=0;
  Udp.begin(localPort);
  sendNTPpacket(timeServer);
  delay(1000);
  if (Udp.parsePacket()){
    Udp.read(packetBuffer,NTP_PACKET_SIZE); // read the packet into the
buffer
    unsigned long highWord, lowWord, epoch;
    highWord = word(packetBuffer[40], packetBuffer[41]);
    lowWord = word(packetBuffer[42], packetBuffer[43]);
    epoch = highWord << 16 | lowWord;
    epoch = epoch - 2208988800 + timeZoneOffset;
    flag=1;
    setTime(epoch);
    ntpLastUpdate = now();
  }
  return flag;
}
```

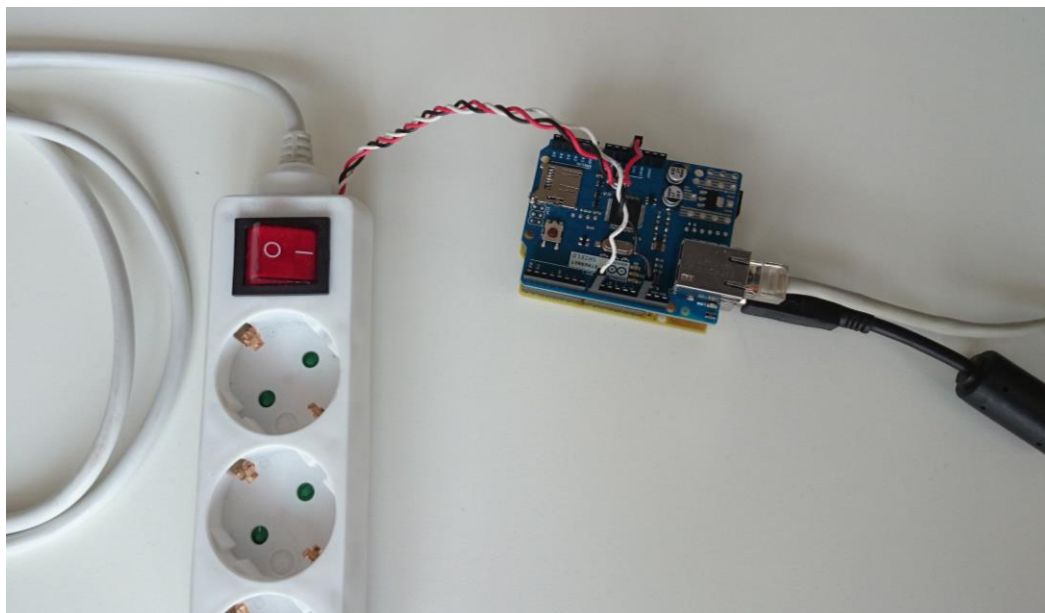
5. Realizacija cijelog sustava

Proces realizacije sustava sigurne i upravljive utičnice sastojao se od planiranja, nabavke potrebnih komponenti te sklopovske i programske realizacije.

Nakon izrade tiskane pločice i njezine ugradnje u kućište produžnog kabela, povezali smo produžni kabel na odgovarajuće pinove Arduina, tj. Ethernet Shielda. Ethernet Shield smo zatim pomoću Ethernet kabela povezali na lokalnu mrežu i time ostvarili internetsku vezu.

Arduino smo zatim USB kabelom spojili na računalo, čime mu osiguravamo napajanje te na taj način učitavamo željeni programski kod na Arduino.

Kada je ostvarena internetska veza, na IP adresi 192.168.0.178 moguće je vidjeti web aplikaciju te odabirom željene upcije upravljati stanjem sigurne i upravljive utičnice.



Slika 7. Prikaz sustava pametne utičnice

6. Zaključak

Razvijeni sustav sigurne i upravljive utičnice omogućuje upravljanje stanjem utičnice na daljinu pomoću web aplikacije. Upravljanje stanjem utičnice, nudi mogućnost upravljanja i radom bilo kojeg trošila spojenog na utičnicu, čime se otvara cijeli niz mogućih primjena.

Sustav pametne utičnice je primjenjiv u svakom kućanstvu koje ima internetsku vezu te ga je moguće koristiti sa svakog uređaja koji se može povezati na Internet. Korist od sustava mogu imati svi koji uz relativno malo uloženi sredstava žele u svom domu implementirati novu funkcionalnost upravljanja utičnicama i trošilima na daljinu ili u unaprijed definirano vrijeme. Sustav ne nudi navedene prednosti onim korisnicima koji ne mogu ostvariti internetsku vezu ili se pak ne znaju koristiti računalom ili pametnim telefonom.

Trenutna ograničenja sustava su u relativno nepouzdanom scheduleru koji zahtjeva stalnu povezanost Arduina s internetom te koji prilikom čitanja podataka o trenutnom vremenu sa servera ne uspijeva uvijek uspostaviti trenutnu vezu. U tom kontekstu, moguće je unaprjeđenje sustava kupnjom RTC modula za Arduino, kao i razvoj kućišta koje bi omogućilo pretvaranje svake zidne utičnice u pametnu utičnicu (umjesto upotrebe produžnog kabela kao što je to slučaj sada).

7. Literatura

- [1] Arduino
<http://en.wikipedia.org/wiki/Arduino> (2015-05-04)
- [2] Instructables – Connecting a 12V Relay to Arduino (2015-05-04)
<http://www.instructables.com/id/Connecting-a-12V-Relay-to-Arduino/>
- [3] PJRC – Time library (2015-05-15)
http://www.pjrc.com/teensy/td_libs_Time.html
- [4] PJRC – Time Alarms library (2015-05-15)
https://www.pjrc.com/teensy/td_libs_TimeAlarms.html
- [5] Instructables – Arduino Internet Time Client (2015-05-15)
<http://www.instructables.com/id/Arduino-Internet-Time-Client/>
- [6] Cyphar – Making a simple scheduler for Arduino ((2015-05-22)
<https://www.cyphar.com/blog/post/making-a-simple-scheduler-for-arduino>
- [7] http://pametne-kuce.zesoi.fer.hr/doku.php?id=2014:popis_projekata (2015-05-04)

8. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Arduino	Razvojna pločica s mikrokontrolerom	http://www.arduino.cc/
Ethernet	Protokol na razini podatkovnog sloja	http://en.wikipedia.org/wiki/Ethernet
MOSFET	Vrsta tranzistora (elektronička sklopka)	http://en.wikipedia.org/wiki/MOSFET
Relej	Vrsta prekidača čije je stanje upravljano posredstvom elektromagneta	http://en.wikipedia.org/wiki/Relay
Scheduler	Svojstvo aplikacije koje nudi definiranje vremena u kojem će se izvršiti određena naredba	https://www.cyphar.com/blog/post/making-a-simple-scheduler-for-arduino