



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

# Pahuljica



- Δ Projekt "Pametna kuća"
- Δ Temperaturni senzor, komunikacija putem Interneta
- Δ Infracrvena komunikacija
- Δ Arduino server

## Sažetak

"Pahuljica" je sustav koji omogućuje uvid u trenutnu temperaturu zraka u prostoriji u kojoj se nalazi te infracrvenu kontrolu nad klimatizacijskim uređajem, koja uključuje paljenje/gašenje i podešavanje temperature. Sve to je realizirano komunikacijom putem Interneta, što omogućuje pristup podacima i kontrolu s bilo kojeg mjesta gdje je dostupna internetska veza. Sustav omogućuje korisnicima stalnu kontrolu nad temperaturom zraka kako bi u svakom trenutku mogli stvoriti ugodni ambijent te optimizirati potrošnju energije. Glavni nedostatak ovog pristupa je upravo ovisnost o internetskoj vezi, međutim, poznavajući današnje okolnosti, taj nedostatak je praktički zanemariv.

## Sadržaj

1. UVOD.....	3
2. DIJELOVI I OPIS SUSTAVA.....	4
2.1. Arduino mikrokontroler .....	4
2.1.1. Arduino web server.....	5
2.1.2. Arduino Ethernet Shield.....	5
2.2. Digitalni termometar .....	6
2.3. Infracrvena komunikacija .....	8
2.3.1. Dekodiranje .....	8
2.3.2. Slanje naredbe.....	9
3. POVEZIVANJE DIJELOVA SUSTAVA.....	9
4. ZAKLJUČAK.....	15
5. LITERATURA.....	16
6. 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 djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

## 1. Uvod

Pojam "pametna" kuća se u posljednje vrijeme sve češće spominje te je pitanje po čemu je ona pametna. Jedno od objašnjenja koje se nameće je da je to automatizirana kuća koja radi neke jednostavne zadatke koji se često ponavljaju ili ih je potrebno često ažurirati. Budući da u današnjem užurbanom svijetu nitko nema previše vremena pratiti sve procese koji se odvijaju u kući i izvan nje, logičan je potez prepustiti neke zadatke da se brinu sami o sebi. "Pahuljica" je samo jedan dio cjelokupnog sustava koji čini pametnu kuću, a omogućuje korisnicima upravljanje klimatizacijskim uređajem s bilo koje lokacije koja nudi pristup Internetu.

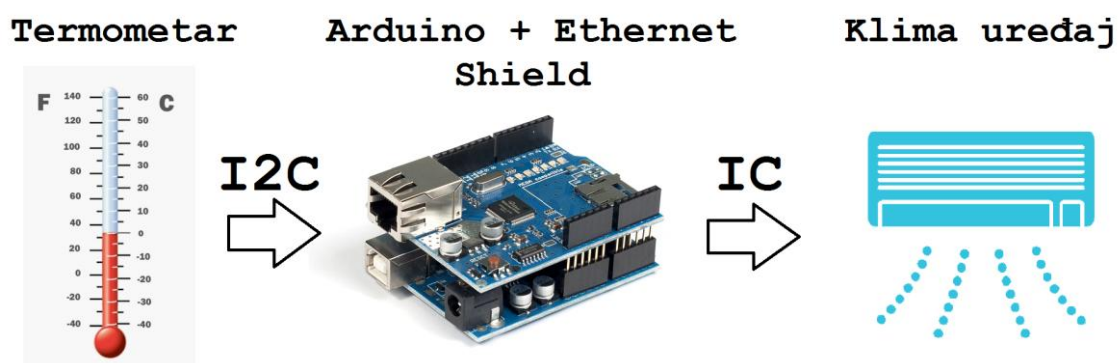
Na prvi pogled svrha ovog sustava se može činiti nepotrebna i banalna, međutim potrebno je stvar sagledati iz druge perspektive. Primarna funkcija sustava je s bilo koje lokacije i u bilo koje vrijeme pratiti temperaturu zraka u svom domu te, ukoliko je temperatura previsoka, a želja nam jer vratiti se u ugodno rashlađen prostor, podesiti željenu temperaturu zraka. Međutim, brojni klimatizacijski uređaji imaju i mogućnost grijanja, čime je uključena mogućnost grijanja prostora u hladnim zimskim danima. Obje ove funkcionalnosti mogu dovesti do značajnije štednje električne energije jer pružaju planiranu potrošnju. U krajnjem slučaju, budući da sustav pruža mogućnost praćenja temperature zraka, moguće je uočiti i određene anomalije poput previsoke temperature (npr. požar) ili preniske temperature (npr. razbijen prozor u zimskoj mećavi).

Sustav je vrlo jednostavno izveden: povezivanjem Arduino mikrokontrolera na Internet pomoću Ethernet modula omogućuje se dvostruka uloga - Arduino se ponaša kao server (postavlja web stranicu, ispisuje trenutne podatke na nju te prima podatke preko Interneta) te kao klasičan mikrokontroler gdje, ovisno o primljenim podacima, upravlja sustavom na koji je spojen. Komunikacija s klimatizacijskim uređajem je izvedena pomoću infracrvene LE diode, a podaci o temperaturi se dobivaju uz pomoć digitalnog termometra DS1621, koji je s Arduinom povezan putem I<sup>2</sup>C veze.

Alternativa opisanom sustavu bi eventualno bio neki komercijalni sustav koji omogućuje istu funkcionalnost, ali budući da se većina potrebnih komponenti može kupiti po vrlo niskoj cijeni i relativno jednostavno spojiti, pitanje je koliko bi alternativa bila isplativa.

## 2. Dijelovi i opis sustava

Sustav se sastoji od Arduino mikrokontrolera, Ethernet shilda koji omogućuje spajanje na Internet, digitalnog termometra/termostata DS1621 za prikupljanje informacija o temperaturi te IC odašiljačke diode za upravljanje kontrolnih signala klimatizacijskom uređaju.



Slika 1: Shema sustava

### 2.1. Arduino mikrokontroler

Arduino je naziv za izvorno talijanske mikrokontrolere koji na jednoj pločici imaju mikrokontroler (najčešće Atmel AVR), pripadno napajanje, USB priključak (za napajanje i programiranje), odvojen priključak za DC napajanje, programator za jednostavno višestruko programiranje, kristalni oscilator, 6 analognih ulaza, 14 digitalnih ulazno-izlaznih pinova te mogućnost spajanja velikog broja modula (Ethernet, Xbee, RFID i slično). Arduino je nastao 2005. godine od strane studenata talijanskog sveučilišta u Ivrei zbog prevelike cijene sustava koji su morali koristiti (BASIC Stamp). Sustav je otvorenog koda, što je potaklo brojne proizvođače na izradu vlastitih kopija, takozvanih "Arduino klonova". Danas se mogu naći kineske kopije koje imaju jednake funkcionalnosti kao i izvorni Arduino, a koštaju daleko manje. Postoji više vrsta Arduino mikrokontrolera, kao što su Uno, Deumilanove, Diecimila, Nano, Leonardo, Mega, LilyPad te Due (koji koristi ARM mikrokontroler). Svaki od njih se odlikuje određenim brojem I/O pinova, količinom memorije, veličinom itd. [1] [2]

Budući da je prvotno zamišljen kao mikrokontroler koji će omogućiti svakome da iskusi programiranje na razini hardvera, korišten je i prilikom razvoja studentskih projekata pametne kuće u sklopu ovog kolegija. U sklopu sustava "Pahuljica" korišten je Arduino Uno, koji služi kao jezgra cijelog sustava jer se na njega spajaju sve komponente te se podaci u

sustavu razmjenjuju isključivo preko njega. Tako Arduino istovremeno prikuplja podatke o temperaturi s digitalnog termometra DS1621 preko I<sup>2</sup>C veze, poslužuje server na kojemu objavljuje podatke o temperaturi te prima komande koje je potrebno putem IC LE diode poslati klimatizacijskom uređaju. Programiranje mikrokontrolera se jednostavno vrši pomoću osobnog računala, pripadnog programa za pisanje programskog koda i USB priključka za slanje programa u memoriju mikrokontrolera.



**Slika 2: Arduino Uno mikrokontroler**

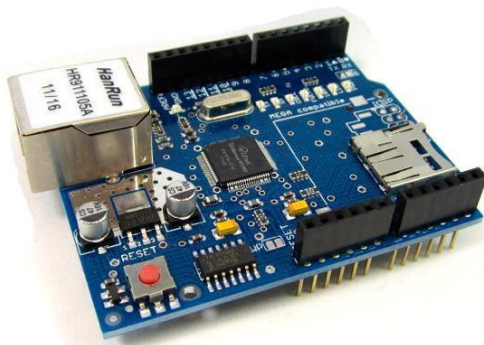
### 2.1.1. Arduino web server

Sustav poslužuje server na samom mikrokontroleru uz pomoć navedenog Ethernet modula, koji se s mikrokontrolerom spaja putem SPI sabirnice, a s ruterom putem UTP kabela. Korištena je *open-source* biblioteka Ethernet.h koja omogućuje da mikrokontroler bude poslužitelj (server) ili klijent. Cijeli kod web stranice i posluživanja zahtjeva se jednostavnim naredbama "upiše" u programski kod mikrokontrolera. Za kreiranje stranice i funkcionalnosti servera korišteni su HTML, CSS, JavaScript, AJAX te XML. Upravo AJAX i XML omogućuju fluidno i diskretno dohvaćanje podataka o temperaturi koje eliminira potrebu konstantnog osvježavanja stranice radi prikaza novih podataka.

### 2.1.2. Arduino Ethernet Shield

Arduino Ethernet shield omogućuje mikrokontroleru spajanje na Internet pomoću klasičnog UTP (RJ-45) kabela. Shield se temelji na Wiznet W5100 ethernet čipu koji podržava i TCP i UDP protokol. Jednostavno se spaja na mikrokontroler putem postojećih pinova na

bočnim stranama mikrokontrolera, odakle crpi potrebno napajanje, uspostavlja vezu s mikrokontrolerom preko SPI sabirnice te uz sve to zadržava postojeći raspored pinova i dozvoljava uobičajeno spajanje periferije na odgovarajuće pinove. Uz to, pozitivna strana shilda je podrška za SD micro memorijsku karticu, koja uvelike može smanjiti potrošnju ograničene memorije na samom mikrokontroleru. Najčešće se na SD karticu sprema web stranica koju mikrokontroler poslužuje, ali moguće je spremati i druge podatke koji bi nepotrebno zauzimali memoriju, a do kojih se lako pristupa koristeći pripadnu SD.h biblioteku.



Slika 3: Ethernet shield

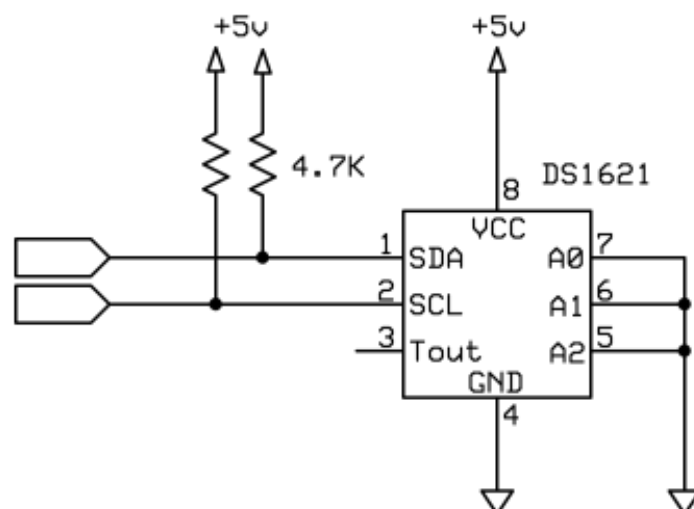
## 2.2. Digitalni termometar

Kao senzor temperature korišten je digitalni termometar/termostat DS1621 tvrtke Maxim. Radi na naponu napajanja između 2,7V i 5,5V te se napaja izravno iz napajanja mikrokontrolera od 5V. U sklopu ovog projekta korištena je samo funkcionalnost termometra jer regulaciju temperature vrši sam mikrokontroler. Senzor se na mikrokontroler priključuje dvožičnom I<sup>2</sup>C vezom (danas se može pronaći i pod nazivom *Two-Wire*) koristeći samo 2 pina, SDA (prijenos podataka) i SCL (sinkronizacijski takt od mikrokontrolera). Budući da su pinovi kod ovog načina vezanja u stanju otvorenog odvoda (*drain*), potrebno je osigurati pull-up otpornike prema napajanju na svakoj od žica. Odabrani su otpornici od 4.7kΩ, kako je preporučeno u datasheetu. Postoje brojni programski kodovi koji objašnjavaju kako konfigurirati termometar da daje temperaturu u željenom formatu te je u ovom projektu korištena besplatna biblioteka DS1621.h [6] jer u svega 3 linije koda pruža jednaku funkcionalnost kao i nepregledna inicijalizacija *Two-Wire* komunikacije.

Također, na ovaj način je moguć prikaz temperature na dvije decimale, za razliku od inicijalne razlučivosti od  $0.5^{\circ}\text{C}$ . Temperatura se u mikrokontroler sprema kao cijelobrojna varijabla od 4 znamenke, te je potrebno ručno izraziti cijeli i decimalni dio temperature (za cijeli dio je potrebno temperaturu podijeliti sa 100, a za decimalni se gleda ostatak pri modulo dijeljenju sa 100).



Slika 4: DS1621 čip



Slika 5: Način spajanja termometra

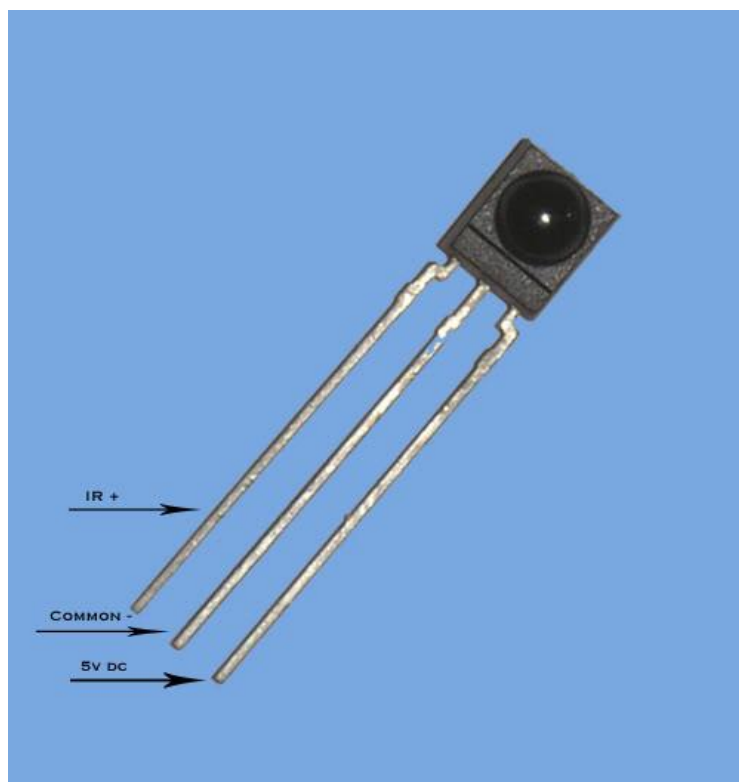


### 2.3. Infracrvena komunikacija

Komunikacija mikrokontrolera i klimatizacijskog uređaja se odvija infracrvenim putem i sastoji se od dva glavna koraka: dekodiranja infracrvenih signala iz daljinskog upravljača te slanja dekodiranih signala u smjeru uređaja.

#### 2.3.1. Dekodiranje

Da bi se moglo komunicirati s nekim uređajem infracrvenim putem, potrebno je poznavati koje signale daljinski upravljač odašilje pritiskom na određenu tipku. Dekodiranje tih signala se jednostavno provodi korištenjem infracrvene prijemne diode. U ovom slučaju korištena je TSOP4838 dioda standardne frekvencije 38kHz. Dioda ima 3 pina, od kojih se 2 spajaju na masu i napajanje, a treći na jedan od digitalnih I/O pinova mikrokontrolera (slika 6). Dekodiranje se vrši uz pomoć *open-source* biblioteke IRremote.h [7] i već napisanog programa IRdecode, čime se dobiva takozvani "sirovi" (*raw*) format odaslanog signala, odnosno trajanje visokih i niskih razina diode pri odašiljanju.



Slika 6: Infracrvena prijemna dioda



### 2.3.2. Slanje naredbe

Nakon što su dekodirani infracrveni kodovi svih tipki koje će se koristiti, potrebno je spremiti kodove u memoriju mikrokontrolera kao polje cijelobrojnih vrijednosti. Kada se pojavi zahtjev za slanje određenog koda (koji se dobiva na web stranici koju poslužuje mikrokontroler), poziva se funkcija IRsend, koja se također nalazi u ranije spomenutoj IRremote.h biblioteci. Izvorni kod u biblioteci predviđa spajanje pozitivnog pina odašiljačke IC diode na treći digitalni pin mikrokontrolera, a negativni pin diode se spaja na GND. Diodu je potrebno usmjeriti prema klimi.



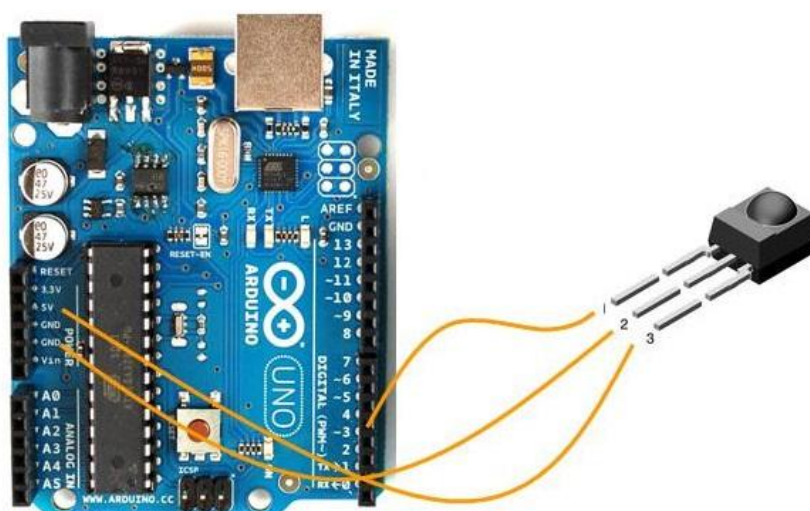
**Slika 7: Infracrvena odašiljačka dioda**

## 3. Povezivanje dijelova sustava

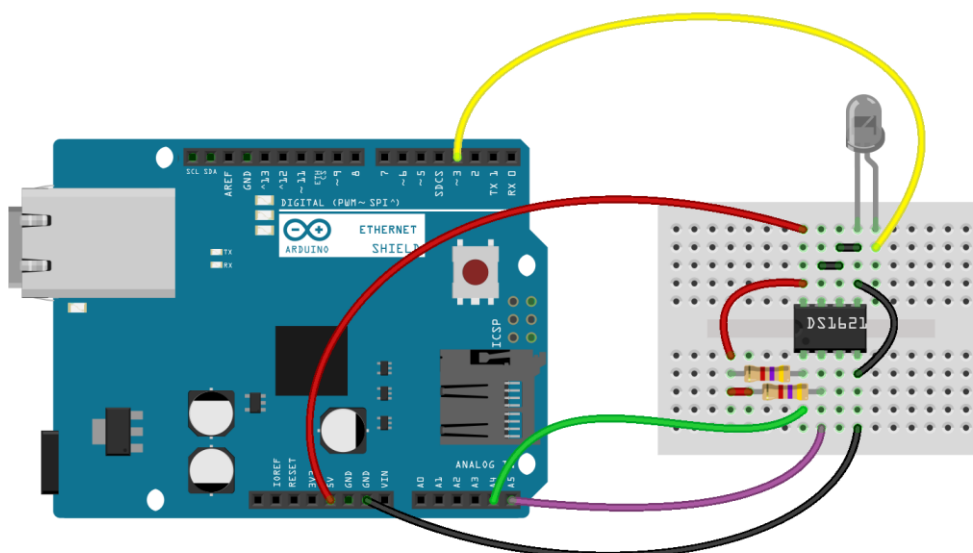
Nakon što su dekodirane sve potrebne tipke daljinskog upravljača i kodovi spremljeni u memoriju mikrokontrolera, potrebno je povezati cijeli sustav u cjelinu. Povezivanje pojedinih dijelova je opisano u prethodnom poglavlju te će ovdje biti samo rezime cijelog postupka.

Poželjno je prvo spojiti sve komponente na predviđena mjesta i tek onda dovesti napajanje. Prvo se spaja Ethernet shield na mikrokontroler tako da se duge iglice s donje strane pločice umetnu u kompatibilne pinove na Arduino. Termometru DS1621 potrebno je osigurati napajanje na priključke  $V_{DD}$  (pin 8) i GND (pin 4), spajanjem na 5V napajanje koje daje mikrokontroler. Dvožična veza ostvaruje se povezivanjem SDA i SCL pinova (pinovi 1 i 2 na termometru) sa pripadnim SDA i SCL pinovima na mikrokontroleru (analogni pinovi 4 i 5). Ne zaboraviti na pull-up otpornike od 4.7k $\Omega$ . Budući da se ostali pinovi na termometru ne koriste, potrebno

je pinove A0 (pin 7), A1 (pin 6) i A2 (pin 5) spojiti na masu, a izlaz  $T_{out}$  ostaviti u zraku. Odašiljačku IC diodu spojiti na digitalni izlaz 3. Na kraju je potrebno UTP kabelom spojiti Ethernet shield sa ruterom i dovesti napajanje na mikrokontroler (koristeći USB kabel ili DC ulaz). U slučaju nepravilnog rada potrebno je resetirati sustav pritiskom na tipku koja se nalazi na Ethernet shieldu.



**Slika 8: Spajanje infracrvene prijemne diode i mikrokontrolera (dekodiranje)**



**Slika 9: Spajanje termometra, infracrvene odašiljačke diode i mikrokontrolera**

## Programski kod sustava

```
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include <DS1621.h>
#include <IRremote.h>

int tC=0;
char power []= "OFF";

// funkcija za slanje kodova
IRsend irsend;

// kodovi nisu prikazani u cijelosti zbog duljine.
// IR kod za ON/OFF
unsigned int raw_onoff[100] =
{4650,2400,450,350,350,350,400,350,350,350,400,350,400, ...

// IR kodovi za temperature 18 do 25 stupnjeva.
unsigned int raw_18[100] =
{4650,2400,450,300,350,300,450,350,350,300,400,350,400,350, ...
unsigned int raw_19[100] =
{4600,2400,400,350,300,350,400,300,300,350,450,300,450,350, ...
unsigned int raw_20[100] =
{4600,2400,450,300,350,400,350,350,350,450,350,400,300, ...
unsigned int raw_21[100] =
{4650,2400,450,350,350,350,400,350,350,300,450,350,400,350, ...
unsigned int raw_22[100] =
{4650,2400,450,300,300,350,450,350,300,300,400,300,450,300, ...
unsigned int raw_23[100] =
{4650,2400,400,350,350,350,400,300,350,350,400,350,400,350, ...
unsigned int raw_24[100] =
{4600,2400,400,350,300,300,400,300,350,350,400,350,400,300, ...
unsigned int raw_25[100] =
{4650,2400,450,300,350,350,450,350,350,300,450,300,400,350, ...

byte addr = (0x90 >> 1) | 0;
DS1621 sensor=DS1621(addr);

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 5, 90);
EthernetServer server(80);

String HTTP_req;

void setup()
{
  sensor.startConversion(false);
  delay(5);

  Ethernet.begin(mac, ip);
  server.begin();
}

void loop()
{
  tC = sensor.getHrTemp(); // citanje temperature sa DS1621 (visoka
                           rezolucija)
```

```

EthernetClient client = server.available();

if (client) {
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      HTTP_req += c; // spremi HTTP zahtjev
      if (c == '\n' && currentLineIsBlank) {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: keep-alive");
        client.println();
        // ako postoji AJAX zahtjev
        if (HTTP_req.indexOf("read_temp") > -1) {
          // procitaj trenutnu temperaturo i posalji
          getTemp(client);
        }
        else {
          // web stranica
          client.println("<!DOCTYPE html>");
          client.println("<html>");
          client.println("<head>");
          client.println("<style>.dizajn { width: 60px; height:
            30px; font-size:28px; font-weight:bold;
            text-align:center; }");
          client.println(".ON { color: green;}");
          client.println(".OFF { color: red; }");
          client.println("</style>");
          client.println("<title>Pahuljica - kontrola
            klime</title>");
          client.println("<script>");
          client.println("function getData() {}");
          client.println(
            "nocache = \"&nocache=\" + Math.random() *
            1000000;");
          client.println("var request = new XMLHttpRequest();");
          client.println("request.onreadystatechange = function()
            {}");
          client.println("if (this.readyState == 4) {}");
          client.println("if (this.status == 200) {}");
          client.println("if (this.responseText != null) {}");

          client.println("document.getElementById
            (\"temperatura\").
            innerHTML = this.responseText;");
          client.println("}}}}");
          client.println("request.open(\"GET\", \"read_temp\" +
            nocache, true);");
          client.println("request.send(null);");
          client.println("setTimeout('getData()', 1500);");
          client.println("");
          client.println("</script>");
          client.println("</head>");
          client.println("<body onload=\"getData()\">");
          client.println("<h1>Pahuljica</h1><br/>");
          client.println("<div id=\"temperatura\">");
          client.println("<h3>Podesi temperaturo:</h3> <h2>");
          client.println("<form method=\"get\">");
          client.println("<input class=\"dizajn\" type=\"text\"
            maxLength=\"3\" name=\"set_temp\"
            value=\"\" > &degC");

```

```

client.println("<br /><input type=\"button\"
              value=\"Potvrđi\" onclick=\"submit();\">");
client.println("</form>");
client.println("</html>");
client.println("</body>");

// postoji li zahtjev za promjenom temperature?
if (HTTP_req.indexOf("set_temp=") > -1) {
char index = HTTP_req.indexOf("set_temp=");
char a = HTTP_req[index+9];
char b = HTTP_req[index+10];
char c = HTTP_req[index+11];
char val = 10*(a-48) + (b-48); // char -> int

// ON ili OFF zahtjevi
if ((a==79 || a==111) && (b==78 || b==110) && (c==32)) {
    strcpy(power, "ON");
    irsend.sendRaw(raw_onoff,100,38);
}

else if ((a==79 || a==111) && (b==70 || b==102) && (c==70 ||
c==102)) {
    strcpy(power, "OFF");
    irsend.sendRaw(raw_onoff,100,38);
}

// zahtjevi za promjenom temperature

switch (val) {
case 18:
    irsend.sendRaw(raw_18,100,38);
    break;
case 19:
    irsend.sendRaw(raw_19,100,38);
    break;
case 20:
    irsend.sendRaw(raw_20,100,38);
    break;
case 21:
    irsend.sendRaw(raw_21,100,38);
    break;
case 22:
    irsend.sendRaw(raw_22,100,38);
    break;
case 23:
    irsend.sendRaw(raw_23,100,38);
    break;
case 24:
    irsend.sendRaw(raw_24,100,38);
    break;
case 25:
    irsend.sendRaw(raw_25,100,38);
    break;
default:
    NULL;
}

index=0;
}
}
HTTP_req = "";
break;
}

```

```
        if (c == '\n') {
            currentLineIsBlank = true;
        }
        else if (c != '\r') {
            currentLineIsBlank = false;
        }
    } // end if
} // end while
delay(5);
client.stop(); // prekini vezu
} // end if (client)
}

// posalji trenutnu temperaturu na web stranicu
void getTemp(EthernetClient cl) {

    cl.print("<h3>Temperatura:</h3> <h2>");
    cl.print(tC / 100); // dio ispred decimalne tocke
    cl.print(".");
    cl.print(tC % 100); // dio iza decimalne docke
    cl.print("&degC");
    cl.print("</h2>");
    cl.print("<h3 class=\"");
    cl.print(power);
    cl.print("\>Status: ");
    cl.print(power);
    cl.print("</h3><br />");
}
```

## 4. Zaključak

Opisani sustav ("Pahuljica") omogućuje svakoj osobi koja ima pristup stranici koju mikrokontroler poslužuje da regulira rad klimatizacijskog uređaja u svom domu, gdje god se nalazili. Prednost sustava je njegova niska cijena i jednostavan dizajn, dok je nedostatak prevelika složenost za ovakav tip mikrokontrolera, što se može ukloniti korištenjem nekog snažnijeg kontrolera (npr. Raspberry Pi). Ipak, sustav na jednostavan način omogućuje praćenje trenutne temperature zraka u domu, te planiranje rada klimatizacijskog uređaja, što vodi do smanjene potrošnje električne energije te, naravno, postizanje ugodnog ambijenta u svom domu.

Koristi od sustava će imati oni koji žele za malo uloženog novca imati kontrolu nad određenim dijelom uređaja u svom domu dok su odsutni. Klimatizacijski uređaj je samo početak postupka automatizacije i monitoringa raznih uređaja s kojima smo u svakodnevnom doticaju. Uz kontrolu uređaja koji koriste infracrvenu vezu, moguće je uspostaviti i Bluetooth vezu i mnoge druge. Smanjenjem dimenzija i cijena kontrolera, svatko je u mogućnosti skrojiti sustav upravljanja svojim domom po vlastitoj želji. Pametne kuće su nekada bile luksuz i maštarije mnogih, a danas su stvarnost i pristupačnije nego ikada.



## 5. Literatura

- [1] Arduino (Wikipedia).  
URL: <http://en.wikipedia.org/wiki/Arduino> (2014-5-28)
- [2] Arduino.  
URL: <http://www.arduino.cc/> (2014-5-28)
- [3] Arduino Web Server.  
URL: <http://arduino.cc/en/Tutorial/WebServer> (2014-4-14)
- [4] Arduino Ethernet Shield Web Server Tutorial.  
URL: <http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/> (2014-4-15)
- [5] Arduino Forum: Read out of DS1621  
URL: <http://forum.arduino.cc/index.php/topic,2162.0.html> (2014-5-3)
- [6] DS1621 temperature probe library for Arduino.  
URL: <https://github.com/martinhandsk/DS1621-temperature-probe-library-for-Arduino/blob/master/DS1621.h> (2014-5-3)
- [7] Arduino IRremote.  
URL: <https://github.com/shirriff/Arduino-IRremote> (2014-5-2)

## 6. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Arduino	Elektronička pločica s mikrokontrolerom i popratnim hardverom. Služi za brzo i jednostavno programiranje prototipa.	<a href="http://www.arduino.cc/">http://www.arduino.cc/</a>
IC (IR)	Infracrvena ( <i>infrared</i> ) komunikacija.	<a href="http://en.wikipedia.org/wiki/Infrared">http://en.wikipedia.org/wiki/Infrared</a>
I <sup>2</sup> C	<i>Inter-integrated Circuit</i> - sabirnica koja služi za povezivanje periferija niske brzine rada.	<a href="http://en.wikipedia.org/wiki/I2c">http://en.wikipedia.org/wiki/I2c</a>
SPI	Serijsko periferijsko sučelje ( <i>interface</i> ).	<a href="http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus">http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus</a>
Ethernet	Najraširenija podvrsta IEEE 802.3 standarda.	<a href="http://en.wikipedia.org/wiki/Ethernet">http://en.wikipedia.org/wiki/Ethernet</a>
UTP kabel	Neoklopljena upletena parica ( <i>Unshielded Twisted Pair</i> ).	<a href="http://en.wikipedia.org/wiki/Twisted_pair">http://en.wikipedia.org/wiki/Twisted_pair</a>
HTML	Jezik za označavanje teksta, koristi se kod izrade web stranica.	<a href="http://en.wikipedia.org/wiki/Html">http://en.wikipedia.org/wiki/Html</a>
CSS	Jezik za opis izgleda i formatiranje web stranica.	<a href="http://en.wikipedia.org/wiki/Css">http://en.wikipedia.org/wiki/Css</a>
JavaScript	Dinamički programski jezik koji nalazi primjene u izradi web stranica ili raznih aplikacija.	<a href="http://en.wikipedia.org/wiki/JavaScript">http://en.wikipedia.org/wiki/JavaScript</a>
AJAX	<i>Asynchronous JavaScript and XML</i> - tehnologija za izradu asinkronih web aplikacija	<a href="http://en.wikipedia.org/wiki/Ajax_%28programming%29">http://en.wikipedia.org/wiki/Ajax_%28programming%29</a>
XML	Proširivi jezik za označavanje ( <i>Extensible Markup Language</i> )	<a href="http://en.wikipedia.org/wiki/Xml">http://en.wikipedia.org/wiki/Xml</a>
Open-source	Naziv za besplatno dijeljenje programskog koda i mogućnost uređivanja (poboljšavanja) istoga po vlastitoj želji	<a href="http://en.wikipedia.org/wiki/Open_source">http://en.wikipedia.org/wiki/Open_source</a>
Pull-up otpornik	Otpornik koji osigurava visoku logičku razinu u sklopu u slučaju odspajanja ulaza/izlaza ili stanja visoke impedancije	<a href="http://en.wikipedia.org/wiki/Pull_up_resistor">http://en.wikipedia.org/wiki/Pull_up_resistor</a>
LE dioda (LED)	Dioda koja emitira svjetlost	<a href="http://en.wikipedia.org/wiki/Led">http://en.wikipedia.org/wiki/Led</a>