TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA
Zagreb, Getaldićeva 4

Nastavni predmet:	**Ugradbeni računalni sustavi**
Vježba: 13	**Ethernet komunikacija & IoT**
Cilj vježbe:	Naučiti koristiti Ethernet Shield i pomoću njega prenositi informacije sa senzora na web stranicu, kao i davati naredbe Arduinu preko web stranice

Upute

Sve zadatke spremi na USB, a u bilježnici za sve zadatke napiši:

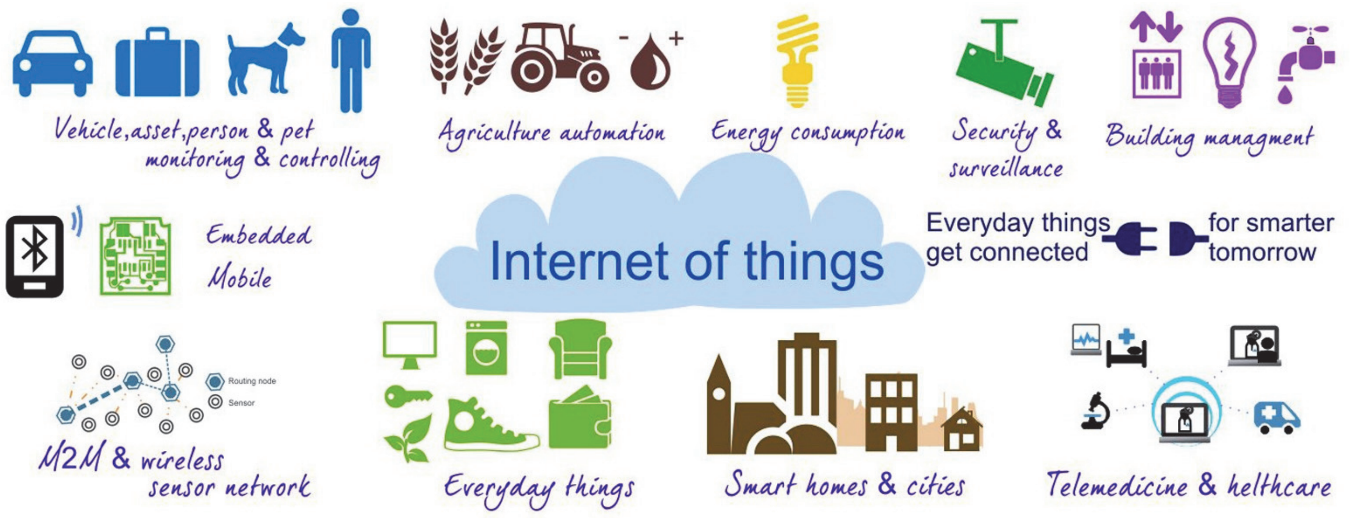
- postupak izrade programa
- objašnjenje korištenih naredbi
- dobivene rezultate po točkama
- odgovoriti u bilježnicu na postavljena pitanja vezana uz ovu vježbu
- Ukoliko u kòdu postoji greška, korigiraj i objasni!

Ethernet komunikacija & IoT

Razvoj informacijske tehnologije - prvenstveno interneta, omogućio je dobivanje informacije u realnom vremenu gotovo iz cijelog svijeta. Jednu takvu mogućnost nam nude web stranice koje daju informacije o temperaturi, vlazi, brzinu vjetra iz svih dijelova svijeta.

Jednostavan sustav takvog tipa možemo konstruirati i sami tako što ćemo primjerice senzor za mjerenje temperature i vlage postaviti u kuću ili u stanu, u učionici, vikendici ili na nekom drugom nama interesantnom mjestu. Ukoliko imamo pristup internetu, te informacije možemo jednostavno pomoću mikroupravljača s Ethernet ili Wifi modulom prenijeti ne web stranicu. Isto tako možemo putem web stranice upravljati funkcijama Arduina.

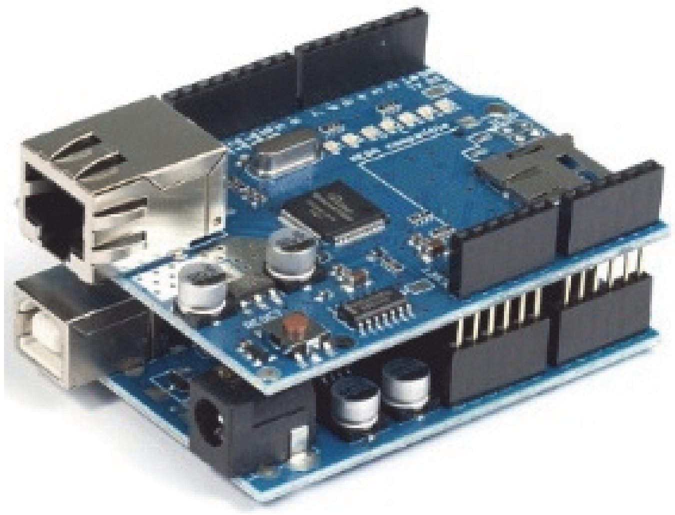
Prava implementacija pojma *Internet of Things*, odnosno Interneta stvari bila bi povezivanje 'stvari' putem interneta, a to možemo ostvariti povezivanjem dva ili više mikroupravljača sa svojim senzorima i aktuatorima i/ili računala preko Ethernet protokola bez obzira na njihovu lokaciju. Internet stvari će se razvijati vrlo brzo budući da se za njegovu implementaciju može koristiti već postojeća infrastruktura.



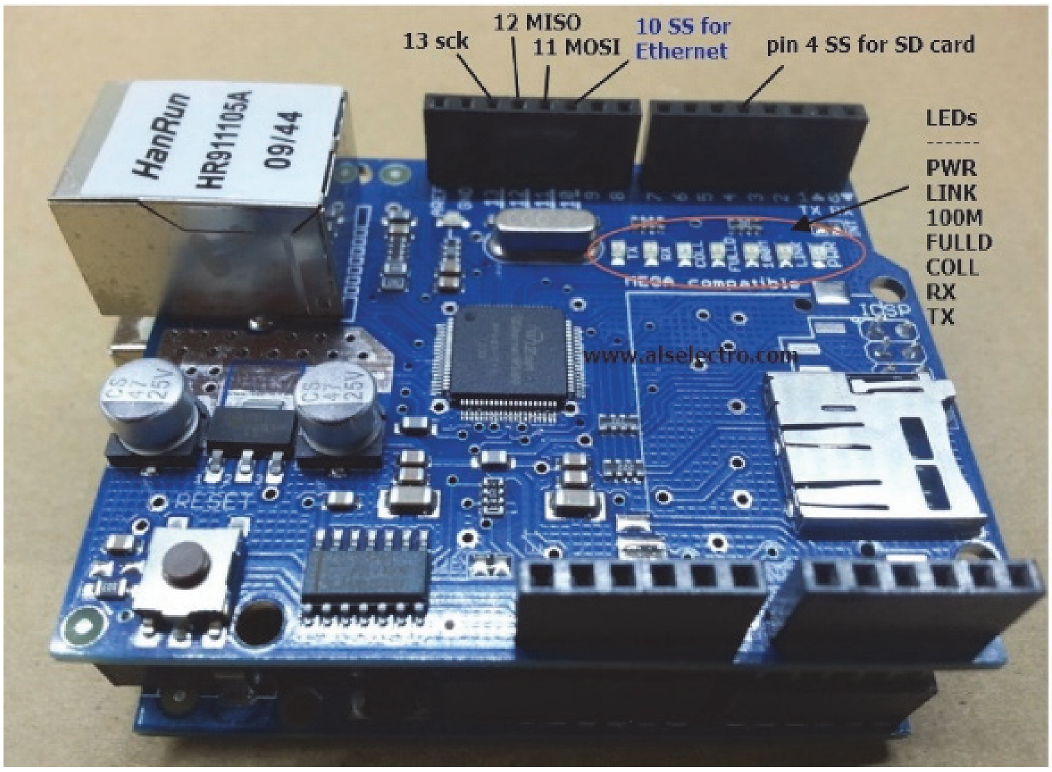
Izvori: www.datasciencebe.com
https://inventrom.wordpress.com/2014/11/27/the-thing-in-internet-of-things/

Na laboratorijskim vježbama koristiti ćemo postojeće senzore i izvršne elemente, a vezu Arduina s internetom i web stranicom ćemo ostvariti pomoću Ethernet shielda.

Ethernet shield koji koristimo na LV se natakne na Arduino UNO (a kompatibilan je i s Arduino Mega). Na taj način nam ostaje raspoloživa većina pinova Arduina kao da shield nije niti priključen.



Na Ethernet shieldu se osim sklopovlja za mrežno povezivanje nalazi i Micro SD utor, koji omogućuje korištenje MicroSD kartice.

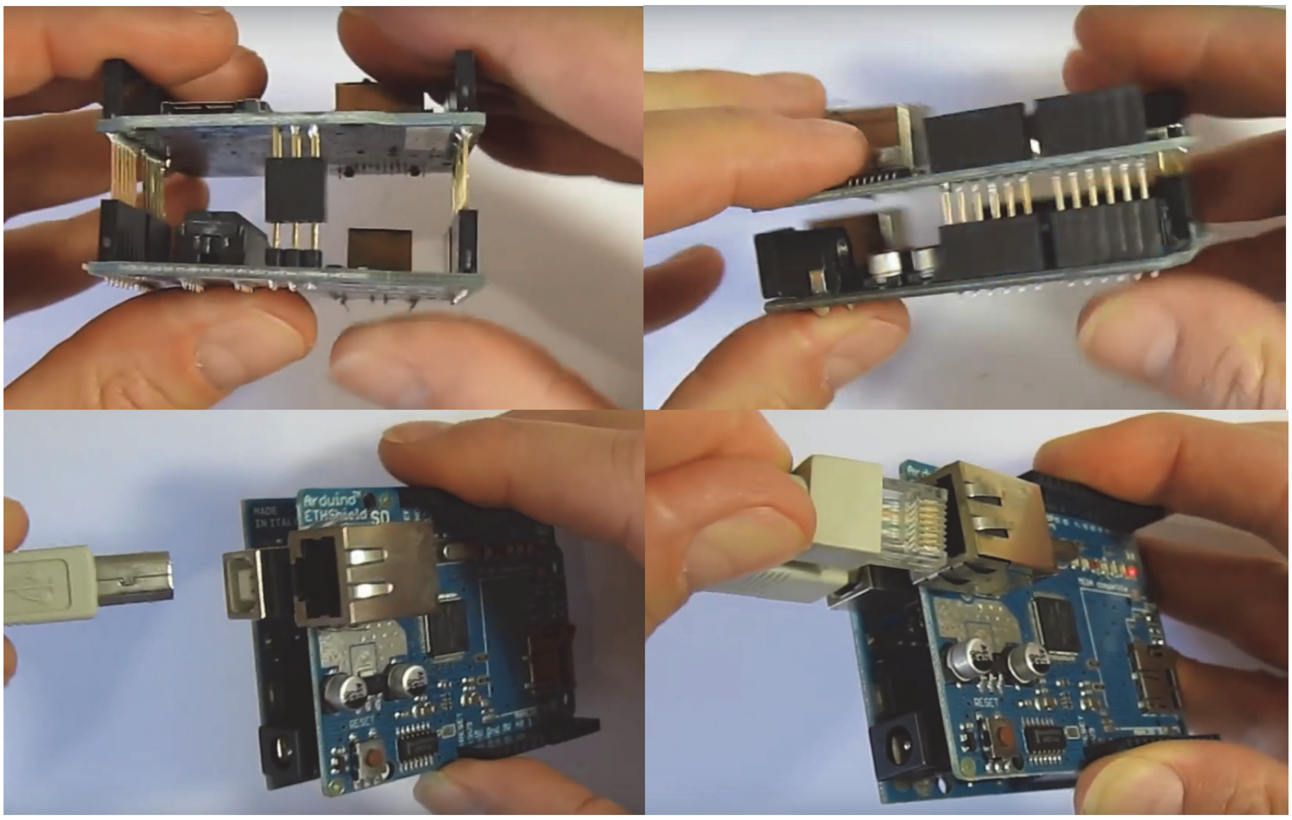


Izvor: http://www.alselectro.com/ethernet-shield.html

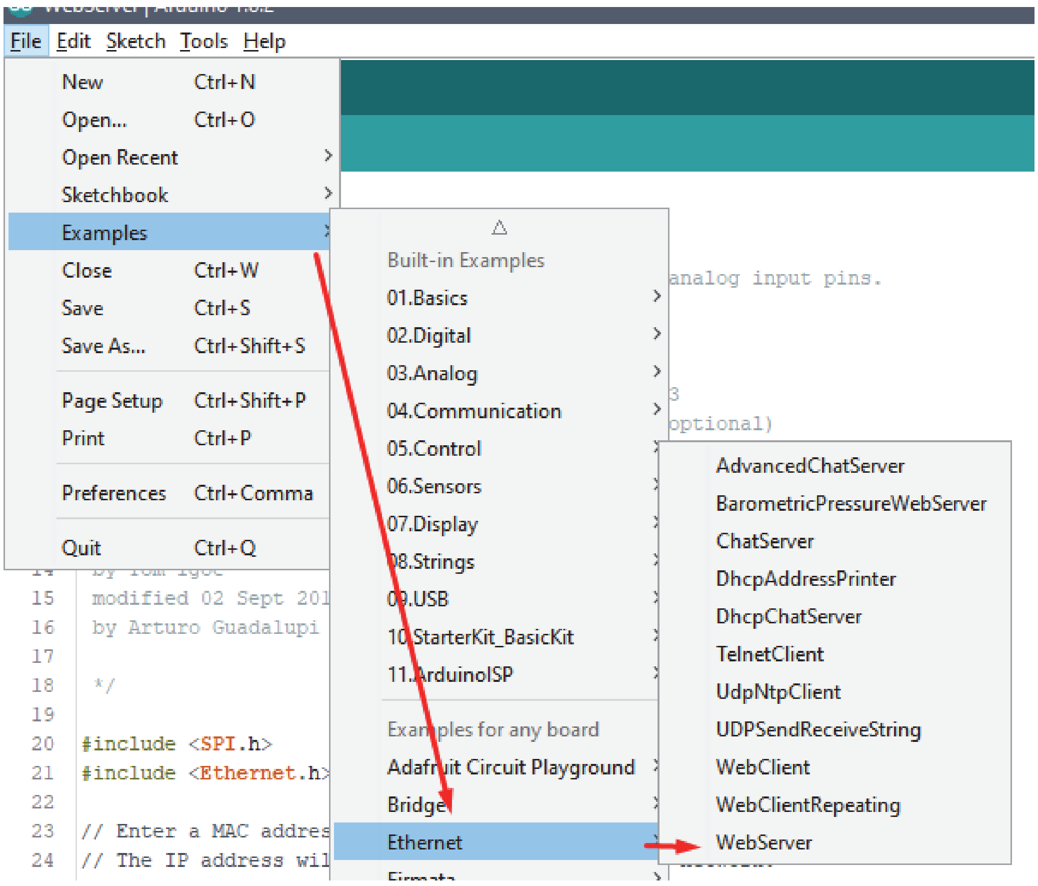
Ethernet shield koristi SPI sabirnicu mikroupravljača ATmega 328p, pa se za komunikaciju između Arduina i Shielda korist pin 11 za signal MOSI, pin 12 za signal MISO i pin 13 za SCK. Za SS linije (*slave select*) kojima se odabire željeni uređaj Ethernet ili SD čitač koriste se pin 10 kao SS linija za Ethernet modul i pin 4 kao SS linija za SD karticu.
Te pinove se ne smije koristiti za ostale primjene, nego treba odabrati neke od ostalih raspoloživih pinova.

<u>Zadatak 1.</u> Spoji spoj prema shemi i učitaj program kojim ćeš provjeriti ispravnost Etherent Shielda tako da napraviš jednostavan WebServer program.

Ethernet shield pažljivo povezati s Arduinom. Pri tome paziti na sve nažice kako bi ušle u odgovarajuće utore i kako ne bi došlo do savijanja nožica Ethernet shielda.



Otvaramo već gotov kod Web server koji se nalazi među primjerima i koji će poslužiti za testiranje.



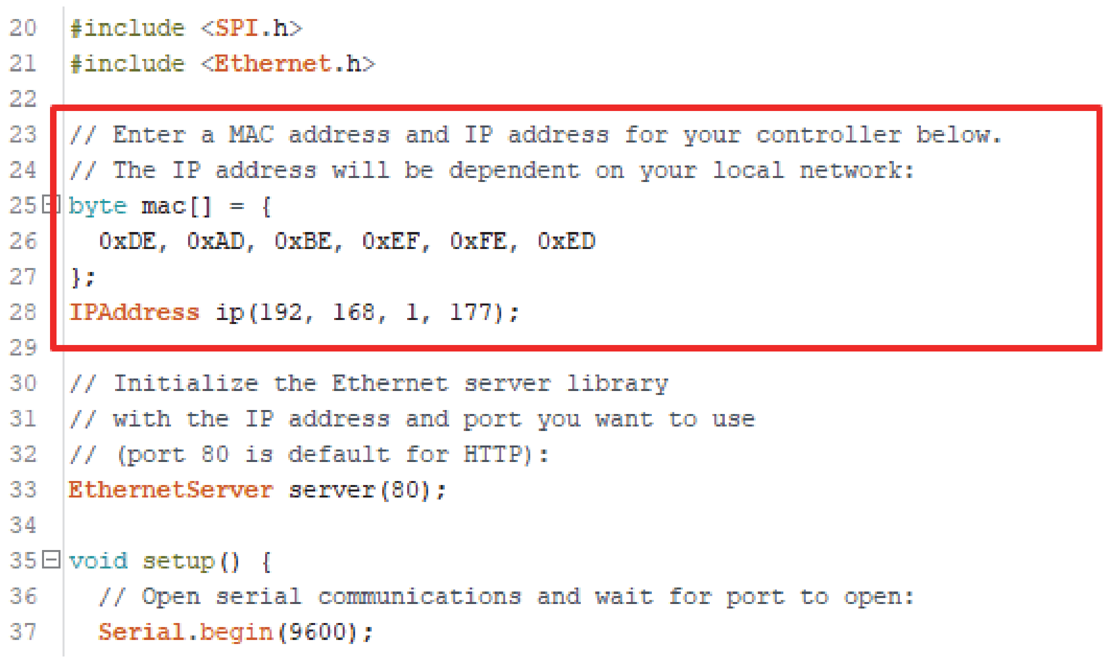
```
analog input pins.
3
optional)
 modified 02 Sept 201
 by Arturo Guadalupi

 */

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

// Enter a MAC addres
// The IP address wil
```

U programu je **potrebno promijeniti IP adresu** u skladu s dogovorenim pravilima kako u laboratoriju ne bi došlo do konflikta IP adresa.

IP adrese treba podesiti u skladu s radnom stanicom na kojoj radite:

za **WS01**, podesiti IP adresu 192.168.**70**.**101**
za **WS02**, podesiti IP adresu 192.168.**70**.**102**, itd.

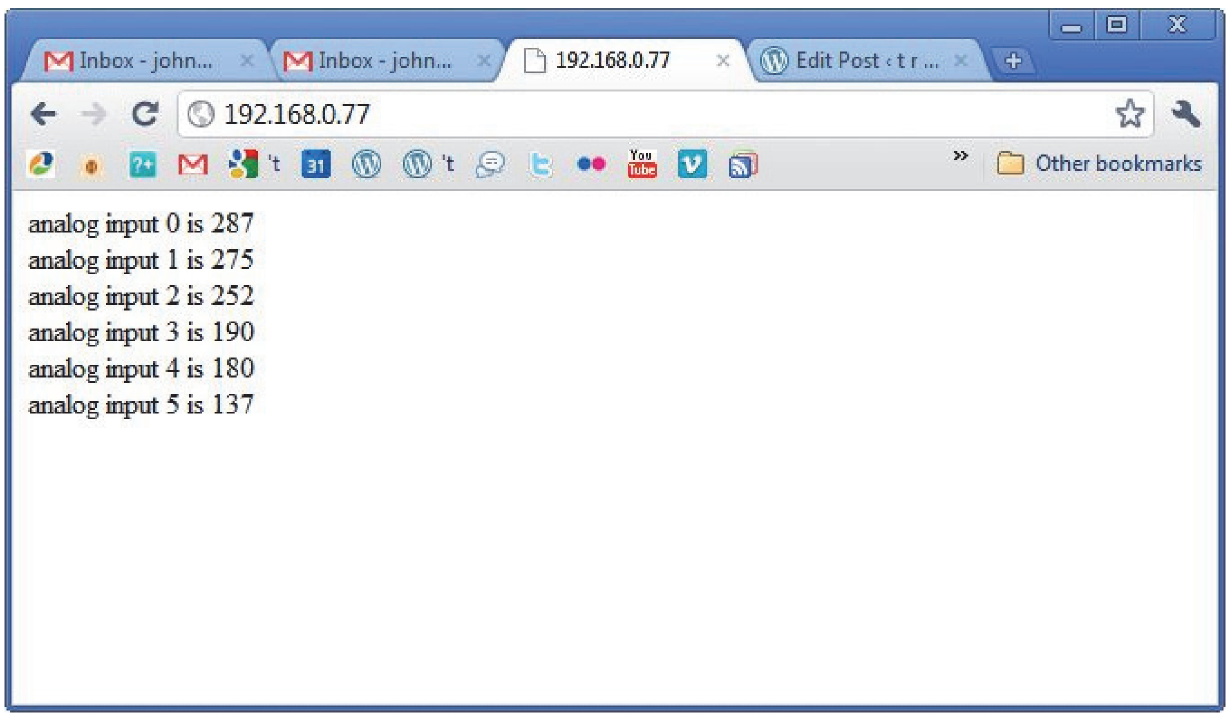
Također je potrebno prepisati MAC adresu s naljepnice na Ethernet shieldu. Ukoliko ga nema, treba odabrati defaultni (što nije preporučljivo) ili iskopirati jedan s interneta (sa random mac generator stranice, npr: https://www.miniwebtool.com/mac-address-generator/).



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

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
```

Nakon uplodanja skice možda će trebati pritisnuti RESET tipku.

Kako bi pristupili svom Ethernet Shieldu upisujemo IP adresu u browser i tamo provjeravamo rezultate.



```
analog input 0 is 287
analog input 1 is 275
analog input 2 is 252
analog input 3 is 190
analog input 4 is 180
analog input 5 is 137
```

Izvor: https://tronixstuff.files.wordpress.com/2010/09/ethernetdemo.jpg

Zadatak 2. Modificiraj prethodni program tako da na web stranici ispišeš „Hello World“. IP i MAC adrese koristiti kao u prvom primjeru.

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

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // PROMIJENITI !!!
IPAddress ip(10, 0, 0, 20); // PROMIJENITI IP ADRESU !!!
EthernetServer server(80);  // create a server at port 80

void setup()
{
    Ethernet.begin(mac, ip);  // initialize Ethernet device
    server.begin();           // start to listen for clients
}

void loop()
{
    EthernetClient client = server.available();  // try to get client

    if (client) {  // got client?
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {      // client data available to read
                char c = client.read();    // read 1 byte (character) from
                                           // client
                // last line of client request is blank and ends with \n
                // respond to client only after last line received
                if (c == '\n' && currentLineIsBlank) {
                    // send a standard http response header
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-Type: text/html");
                    client.println("Connection: close");
                    client.println();
                    // send web page
                    client.println("<!DOCTYPE html>");
                    client.println("<html>");
                    client.println("<head>");
                    client.println("<title>Arduino Web Page</title>");
                    client.println("</head>");
                    client.println("<body>");
                    client.println("<h1>Hello World!</h1>");
                    client.println("</body>");
                    client.println("</html>");
                    break;
                }
                // every line of text received from the client ends with \r\n
                if (c == '\n') {
                    // last character on line of received text
                    // starting new line with next character read
                    currentLineIsBlank = true;
                }
                else if (c != '\r') {
                    // a text character was received from client
                    currentLineIsBlank = false;
                }
            } // end if (client.available())
        } // end while (client.connected())
        delay(1);      // give the web browser time to receive the data
        client.stop(); // close the connection
    } // end if (client)
}
```

Pitanje: Što znači \r ,\n u kodu iznad ?

Kako zapravo radi shield, što je server a što klijent budući da mi upisujemo kod nije baš najlogičnije?

U ovom primjeru, Ethernet Shield radi kao serversko računalo u koje smo programiranjem kreirali web stranicu. Kada upišemo IP adresu Shielda u preglednik na računalu, računalo je zapravo klijent koji sa servera učitava html kod. Klijent šalje serveru slijedeće podatke:

```
GET / HTTP/1.1\r\n
Host: 10.0.0.20\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:17.0) Gecko/20100101 Firefox/17.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-ZA,en-GB;q=0.8,en-US;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
\r\n
```

A server (shield) vraća ako je sve prošlo u redu:

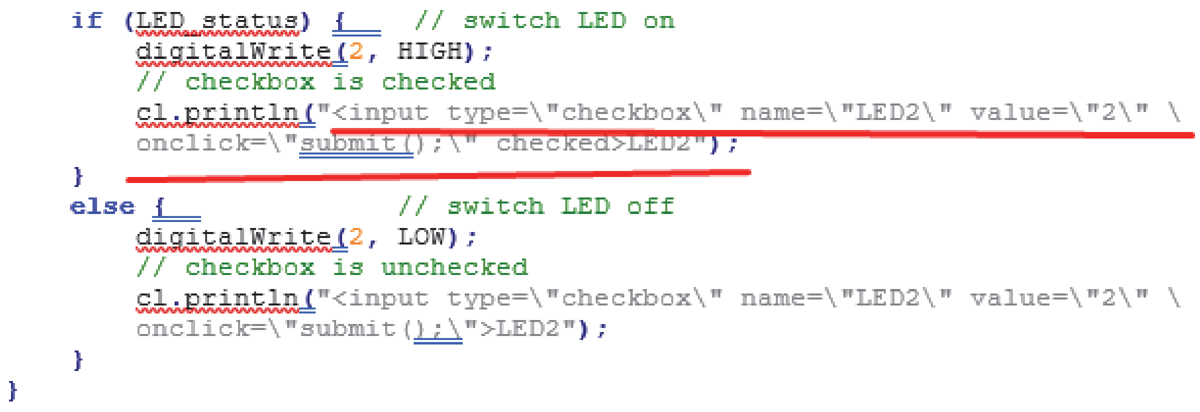
```
HTTP/1.1 200 OK\r\n
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
```

<u>**Zadatak 3.**</u> Modificiraj prethodni program tako da dodaš checkbox na stranicu koja će paliti i gasiti LED diodu.

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

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(10, 0, 0, 20); // PROMIJENITI IP ADRESU !!!
EthernetServer server(80);  // create a server at port 80

String HTTP_req;          // stores the HTTP request
```

```
boolean LED_status = 0;   // state of LED, off by default

void setup()
{
    Ethernet.begin(mac, ip);  // initialize Ethernet device
    server.begin();           // start to listen for clients
    Serial.begin(9600);       // for diagnostics
    pinMode(2, OUTPUT);       // LED on pin 2
}

void loop()
{
    EthernetClient client = server.available();  // try to get client

    if (client) {  // got client?
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {      // client data available to read
                char c = client.read();    // read 1 byte (character) from
                                           // client
                HTTP_req += c;  // save the HTTP request 1 char at a time
                // last line of client request is blank and ends with \n
                // respond to client only after last line received
                if (c == '\n' && currentLineIsBlank) {
                    // send a standard http response header
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-Type: text/html");
                    client.println("Connection: close");
                    client.println();
                    // send web page
                    client.println("<!DOCTYPE html>");
                    client.println("<html>");
                    client.println("<head>");
                    client.println("<title>Arduino LED Control</title>");
                    client.println("</head>");
                    client.println("<body>");
                    client.println("<h1>LED</h1>");
                    client.println("<p>Click to switch LED on and off.</p>");
                    client.println("<form method=\"get\">");
                    ProcessCheckbox(client);
                    client.println("</form>");
                    client.println("</body>");
                    client.println("</html>");
                    Serial.print(HTTP_req);
                    HTTP_req = "";    // finished with request, empty string
                    break;
                }
                // every line of text received from the client ends with \r\n


                    if (c == '\n') {
                    // last character on line of received text
                    // starting new line with next character read
                    currentLineIsBlank = true;
                }
                else if (c != '\r') {
                    // a text character was received from client
                    currentLineIsBlank = false;
                }
            } // end if (client.available())
        } // end while (client.connected())
        delay(1);      // give the web browser time to receive the data
        client.stop(); // close the connection
    } // end if (client)
}

// switch LED and send back HTML for LED checkbox
void ProcessCheckbox(EthernetClient cl)
{
    if (HTTP_req.indexOf("LED2=2") > -1) {  // see if checkbox was clicked
        // the checkbox was clicked, toggle the LED
        if (LED_status) {
            LED_status = 0;
        }
        else {
            LED_status = 1;
        }
    }

    if (LED_status) {    // switch LED on
        digitalWrite(2, HIGH);
        // checkbox is checked
        cl.println("<input type=\"checkbox\" name=\"LED2\" value=\"2\" \
        onclick=\"submit();\" checked>LED2");
    }
    else {              // switch LED off
        digitalWrite(2, LOW);
        // checkbox is unchecked
        cl.println("<input type=\"checkbox\" name=\"LED2\" value=\"2\" \
        onclick=\"submit();\">LED2");
    }
}
```

Pitanje: Kako je u kodu ostvareno paljenje i gašenje led diode? Kako provjerava Arduino je li checkbox pritisnut? Objasni ovu liniju, što se događa?



```
    if (LED_status) {    // switch LED on
        digitalWrite(2, HIGH);
        // checkbox is checked
        cl.println("<input type=\"checkbox\" name=\"LED2\" value=\"2\" \
        onclick=\"submit();\" checked>LED2");
    }
    else {              // switch LED off
        digitalWrite(2, LOW);
        // checkbox is unchecked
        cl.println("<input type=\"checkbox\" name=\"LED2\" value=\"2\" \
        onclick=\"submit();\">LED2");
    }
}
```

Zadatak 4. Modificiraj spoj tako da dodaš potenciometar na pin A0. Prethodni program modificiraj tako da ispišeš A/D vrijednost potenciometra na web stranicu.

Zadatak 5. Modificiraj spoj tako da kreiraš 2 linka na web stranici koji će pozicionirati servo motor u određeni položaj. U jednom položaju neka LED dioda bude upaljena a u drugom ugašena.

```
#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>
Servo myservo;  // create servo object to control a servo

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //physical mac address
byte ip[] = { 192, 168, 0, 14 }; // PROMIJENITI IP ADRESU !!! //subnet mask
EthernetServer server(80);                                 //server port
String readString;

//////////////////////

void setup(){

  pinMode(5, OUTPUT); //pin selected to control
  myservo.attach(7);  //the pin for the servo control
  //start Ethernet
  Ethernet.begin(mac, ip);
  server.begin();

   //enable serial data print
  Serial.begin(9600);
  Serial.println("server servo/pin 5 test 1.0"); // so I can keep track of
what is loaded
}

void loop(){
  // Create a client connection
  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 HTTP request has ended
        if (c == '\n') {
          ///////////////
          Serial.println(readString); //print to serial monitor for debuging

          client.println("HTTP/1.1 200 OK"); //send new page
          client.println("Content-Type: text/html");
          client.println();

          client.println("<HTML>");
          client.println("<HEAD>");
          client.println("<TITLE>Arduino WEB</TITLE>");
          client.println("</HEAD>");
          client.println("<BODY>");


          client.println("<a href=\"/?on\"\">110</a>");
          client.println("<a href=\"/?off\"\">10</a>");

          client.println("</BODY>");
          client.println("</HTML>");

          delay(1);
          //stopping client
          client.stop();

          //////////////////// control arduino pin
          if(readString.indexOf("?on") >0)//checks for on
          {
            myservo.write(10);
            digitalWrite(5, HIGH);    // set pin 4 high
            Serial.println("Led On");
          }
          if(readString.indexOf("?off") >0)//checks for off
          {
            myservo.write(110);
            digitalWrite(5, LOW);    // set pin 4 low
            Serial.println("Led Off");
          }
          //clearing string for next read
          readString="";

        }
      }
    }
  }
}
```

<u>Zadatak 6.</u> Slijedeći zadatak trebaju raditi dvije grupe u paru. Na Arduino 1 spojen je temperaturni senzor DHT 11 ili DHT 22 i LED dioda. Na Arduino 2 je spojen LCD i tipkalo. Potrebno je:

a) Temperaturu izmjerenu na Arduinu 1 ispisati na LCD-u na Arduinu 2
b) Tipkalom s Arduina 2 paliti i gasiti LED na Arduinu 1.

Primjeri koda Arduino 1 koji šalje podatke:

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

// inspired by/copied from http://arduino.cc/en/Tutorial/TelnetClient

// Enter a MAC address and IP address for your controller below:
// The IP address will be dependent on your local network:
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 176);  // this is the data source card IP address

                                                 // the IP address of the server
you're connecting to:
IPAddress server(192.168, 1, 177);

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 23 is default for telnet;
EthernetClient client;
int port = 23;  // telnet default port

char myVar[100]; // contains string with variable to transmit

void setup() {
     // start the Ethernet connection:
     Ethernet.begin(mac, ip);
     // Open serial communications and wait for port to open:
     Serial.begin(9600);
     while (!Serial) {
          ; // wait for serial port to connect. Needed for Leonardo only
     }

     // give the Ethernet shield a second to initialize:
     delay(1000);
     Serial.println("connecting...");

     // if you get a connection, report back via serial:
     if (client.connect(server, port)) {
          Serial.println("connected");
     }
     else {
          // if you didn't get a connection to the server:
          Serial.println("connection failed");
     }
}
```

```
void loop()
{
      // if there are incoming bytes available
      // from the server, read them and print them:
      // the server code above doesn't send anything…
      // but if it did, this is where you would echo it
      int ii;
      if (client.available()) {
            char c = client.read();
            Serial.print("***Server says:***\n");
            Serial.print(c);
      }

      // assume your variable myVar will have a valid string in it...
      strcpy(myVar, "123.456\n");
      // tell the serial port what you are sending:
      Serial.print("sending variable: ");
      Serial.print(myVar);
      for (ii = 0; ii < strlen(myVar); ii++) {
            if (client.connected()) {
                  client.print(myVar[ii]);
            }
      }

      // if the server's disconnected, stop the client:
      if (!client.connected()) {
            Serial.println();
            Serial.println("disconnecting.");
            client.stop();
            // do nothing:
            while (true);
      }
      // add appropriate delay here before sending next data element
}
```

Primjeri koda Arduino 2 koji prima podatke:

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

// network configuration.
// gateway and subnet are optional.

// the media access control (ethernet hardware) address for the shield:
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

//the IP address for the shield:
byte ip[] = { 10, 0, 0, 177 };

// the router's gateway address:
byte gateway[] = { 10, 0, 0, 1 };
```

```
// the subnet:
byte subnet[] = { 255, 255, 0, 0 };

// telnet defaults to port 23
EthernetServer server = EthernetServer(23);

void setup()
{
      // initialize the ethernet device
      Ethernet.begin(mac, ip, gateway, subnet);

      // start listening for clients
      server.begin();
}

void loop()
{
      // if an incoming client connects, there will be bytes available to
read:
      char incoming[100];
      EthernetClient client = server.available();
      if (client == true) {
            // read bytes from the incoming client and write them back
            // to any clients connected to the server:
            int ii = 0;
            while ((c = client.read()) != '\n')
            {
                  incoming[ii++] = c;
            }
            // the variable incoming[] now contains the most recent value sent
            // so you can do something with it
      }
}
```

Literatura:

1. Instructables: Modify the HC-05 Bluetooth Module Defaults Using AT Commands, http://www.instructables.com/id/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/ , (pregledano 07. svibnja 2018.)
2. Instructables: ArduDroid: a Simple 2-Way Bluetooth-based Android Controller for Arduino, http://www.instructables.com/id/Andruino-A-Simple-2-Way-Bluetooth-based-Android-C/ , (pregledano 07. svibnja 2018.)
3. Howtomechatronics: How To Build Custom Android App for your Arduino Project using MIT App Inventor, https://howtomechatronics.com/tutorials/arduino/how-to-build-custom-android-app-for-your-arduino-project-using-mit-app-inventor/ , (pregledano 07. svibnja 2018.)
4. MIT App Inventor, http://appinventor.mit.edu/ , (pregledano 07. svibnja 2018.)