

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 12	Ethernet komunikacija & IoT
Cilj vježbe:	Naučiti koristiti Ethernet Shield i pomoću njega prenositi informacije sa senzora na web stranicu, kao i davati naredbe Arduino preko web stranice

Upute

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

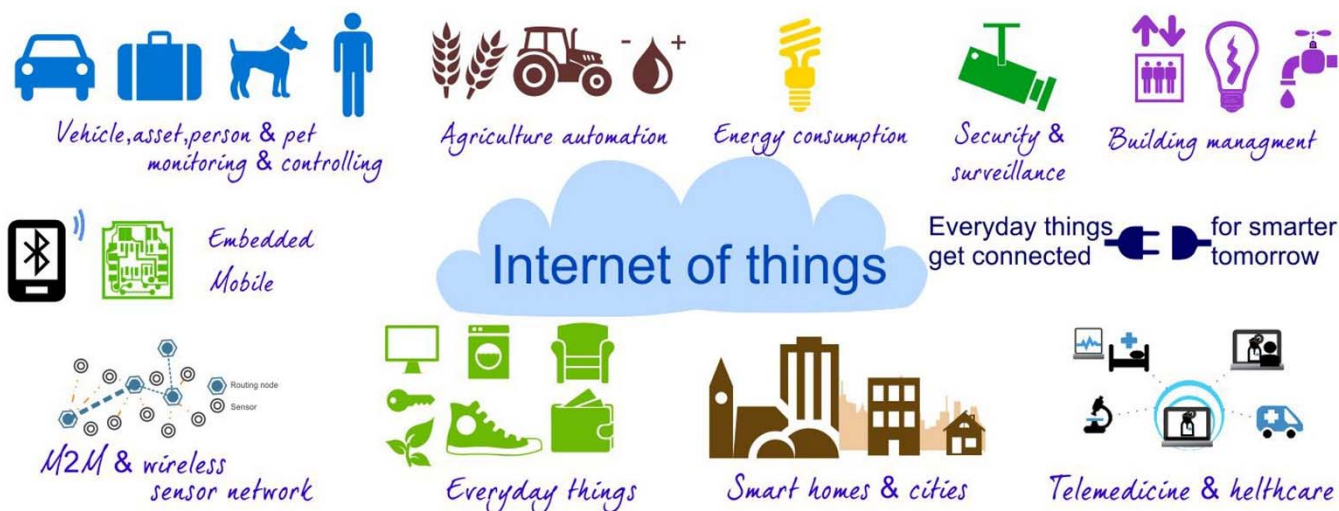
- postupak izrade programa
- objašnjenje korištenih naredbi
- dobivene rezultate po točkama
- odgovoriti u bilježnicu na postavljena pitanja vezana uz ovu vježbu
- Ukoliko u kòdu postoji greška, korigiraj i objasni!

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 na 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 izvršnim elementima (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.



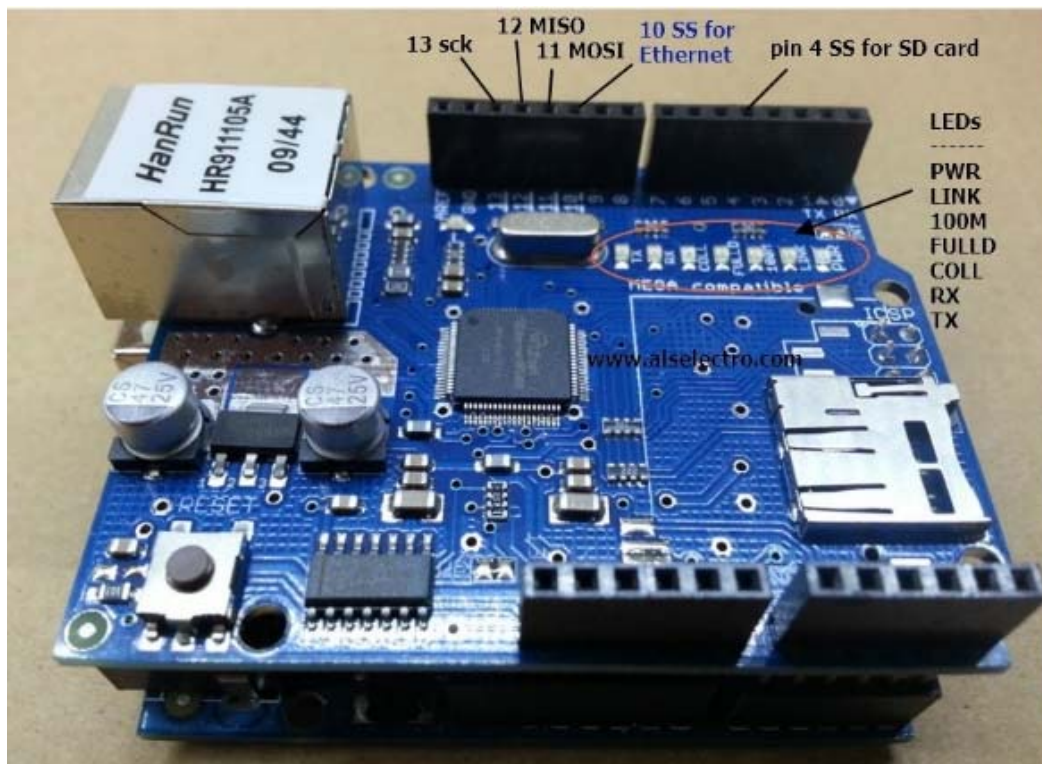
Izvori: www.datasciencebe.com
<https://inventrom.wordpress.com/2014/11/27/the-thing-in-internet-of-things/>

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.

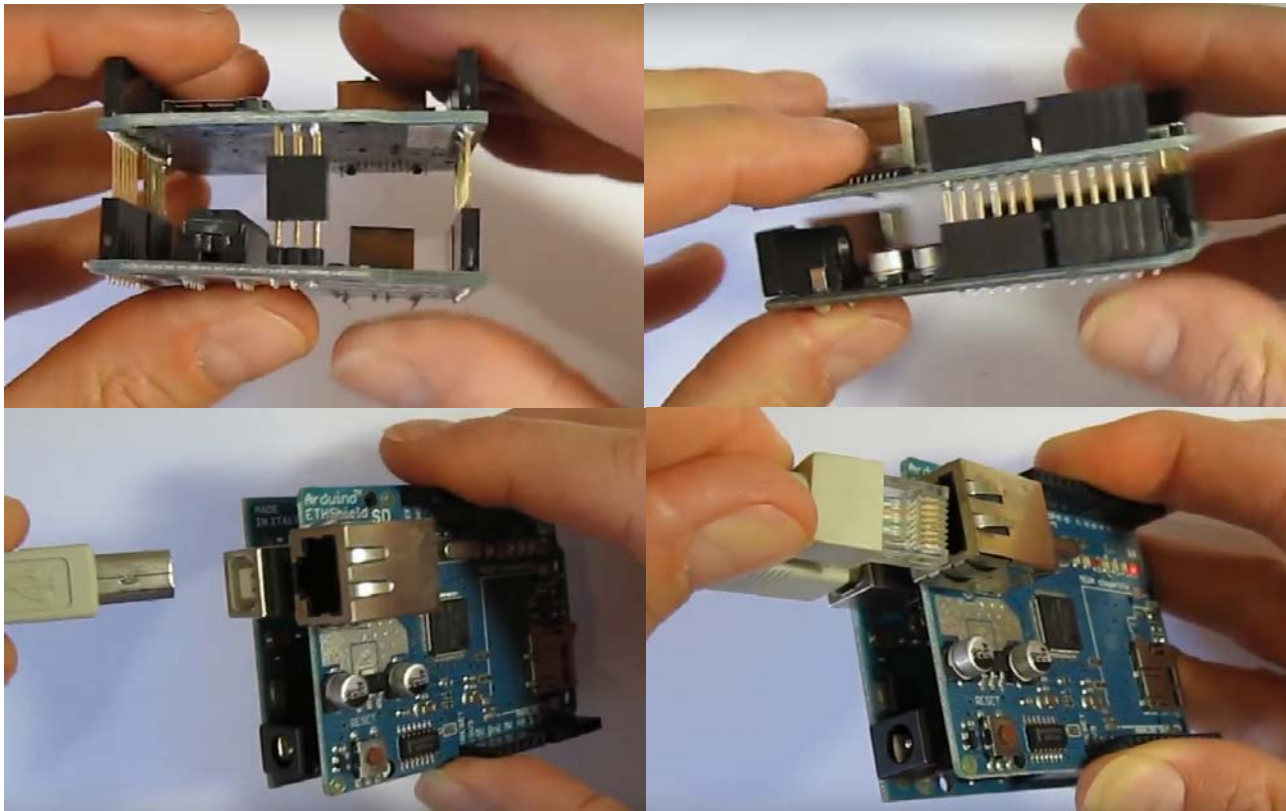


Izvor: <http://www.alselectro.com/ethernet-shield.html>

Ethernet shield koristi SPI sabirnicu mikroupravljača ATmega 328p, pa se za komunikaciju između Arduina i Shielda koriste 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.

Zadatak 1. Ethernet shield pažljivo povezati s Arduinoom. Pri tome paziti na sve nožice kako bi ušle u odgovarajuće utore i kako ne bi došlo do savijanja nožica Ethernet shielda. Unesi programski kod u nastavku kojim ćeš napraviti jednostavan WebServer koji će na klijentsko računalo u web pregledniku ispisivati A/D vrijednosti svih analognih ulaza (bez obzira što na njih nije ništa priključeno).



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

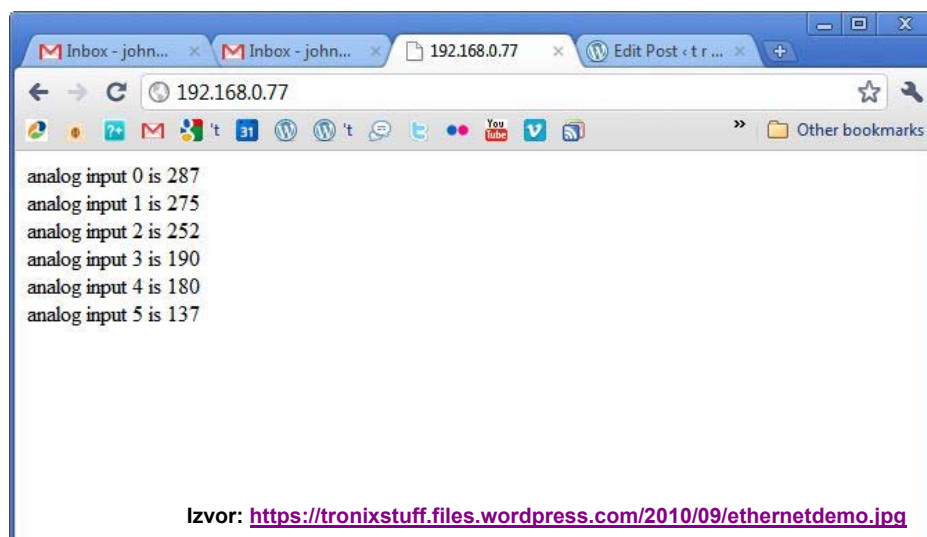
IP adrese treba podesiti u skladu s radnom stanicom na kojoj radite:

za **WS01**, podesiti IP adresu 192.168.**70.101** (192.168.**80.101**)

za **WS02**, podesiti IP adresu 192.168.**70.102**, (192.168.**80.102**)

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/>).

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



Kòd zadatka

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

// http response header kojim će web server odgovoriti klijentu
// ovo će kompajler spojiti u jedan string, ovako je pisano čisto zbog
čitljivosti
const char* http_response_header =
"HTTP/1.1 200 OK\r\n"           // HTTP verzija 1.1, zahtijev je bio uspješan
"Content-Type: text/html\r\n"   // vraćam HTML
"Connection: close\r\n"        // nakon što dobiš podatke, zatvori vezu
"Refresh: 5\r\n"                // ponovno učitaj nakon 5 sekunda
"\r\n";                         // nakon prazne linije slijedi content

// početni i krajni dio web stranice
// između njih se nalazi sadržaj
// ovdje je korištenje "\r\n" opcionalno jer HTML nije briga za whitespace
(razmaci, znak za novu liniju)
const char* html_start =
"<!DOCTYPE html>"
"<html>"
"<body>";

const char* html_end =
"</body>"
"</html>";

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 137, 177);
EthernetServer server(80); // inicijalizira na portu 80 (http)

void setup() {
  Serial.begin(9600);
  while (!Serial); // čekaj da se serial monitor otvori
  Serial.println("\tEthernet webserver");
  Ethernet.begin(mac, ip);

  // ispitivanje je li ethernet shield priključen
  if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    Serial.println("Ethernet shield nije pronađen.");
    while (true); // čekaj zauvijek, jer bez ethernet shielda nema
    potrebe nastavljati kod
  }
  if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet kabel nije priključen.");
  }

  // pokreni server
  server.begin();
  Serial.print("Server je pokrenut na ");
```

```
    Serial.println(Ethernet.localIP());
}

void loop() {
    // čekaj dok se jedan klijent ne spoji
    EthernetClient client = server.available();

    if (client) {
        Serial.println("New client connected:");

        // pročitaj HTTP zahtjev i ispiši na serial monitoru
        while (client.connected() && client.available())
            Serial.print((char)client.read());

        client.print(http_response_header);
        client.print(html_start);

        // ispiši stanje na analognim ulazima
        for (int i = 0; i < 6; i++) {
            int val = analogRead(i);
            client.print("analogni ulaz ");
            client.print(i);
            client.print(": ");
            client.print(val);
            client.println("<br />");
        }

        client.print(html_end);
        client.flush(); // pričekaj dok se sve nije poslalo

        // zatvori vezu
        client.stop();
        Serial.println("client disconnected\n");
    }
}
```

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>

// http response header kojim će web server odgovoriti klijentu
// ovo će kompajler spojiti u jedan string, ovako je pisano čisto zbog
čitljivosti
const char* http_response_header =
    "HTTP/1.1 200 OK\r\n"           // HTTP verzija 1.1, zahtjev je bio uspješan
    "Content-Type: text/html\r\n"  // vraćam HTML
    "Connection: close\r\n"        // nakon što dobiš podatke, zatvori vezu
    "\r\n";                        // nakon prazne linije slijedi content

// potpuni HTML kod web stranice
```

```
const char* html =
"<!DOCTYPE html>"
"<html>"
"<head>"
"<title>Arduino Web Page</title>"
"</head>"
"<body>"
"<h1>Hello, world!</h1>"
"</body>"
"</html>";

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 137, 177);
EthernetServer server(80); // inicijalizira na portu 80 (http)

void setup() {
  Serial.begin(9600);
  while (!Serial); // čekaj da se serial monitor otvori
  Serial.println("\tEthernet webserver");
  Ethernet.begin(mac, ip);

  // ispitivanje je li ethernet shield priključen
  if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    Serial.println("Ethernet shield nije pronađen.");
    while (true); // čekaj zauvijek, jer bez ethernet shielda nema
    potrebe nastavljati kod
  }
  if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet kabel nije priključen.");
  }

  // pokreni server
  server.begin();
  Serial.print("Server je pokrenut na ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // čekaj dok se jedan klijent ne spoji
  EthernetClient client = server.available();

  if (client) {
    Serial.println("New client connected:");

    // pročitaj HTTP zahtjev i ispiši na serial monitoru
    while (client.connected() && client.available())
      Serial.print((char)client.read());

    client.print(http_response_header);
    client.print(html);
    client.flush(); // pričekaj dok se sve nije poslalo
  }
}
```

```
        // zatvori vezu
        client.stop();
        Serial.println("client disconnected\n");
    }
}
```

Zadatak 3. Modificiraj prethodni program tako da dodaš checkbox na stranicu koja će paliti i gasiti LED diodu.

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

#define LED2 2

// http response header kojim će web server odgovoriti klijentu
// ovo će kompajler spojiti u jedan string, ovako je pisano čisto zbog
čitljivosti
const char* http_response_header =
"HTTP/1.1 200 OK\r\n"           // HTTP verzija 1.1, zahtjev je bio uspješan
"Content-Type: text/html\r\n"  // vraćam HTML
"Connection: close\r\n"        // nakon što dobiješ podatke, zatvori vezu
"\r\n";                        // nakon prazne linije slijedi content

// početni i krajni dio web stranice
// između njih se nalazi sadržaj
// ovdje je korištenje "\r\n" opcionalno jer HTML nije briga za whitespace
(razmaci, znak za novu liniju)
const char* html_start =
"<!DOCTYPE html>"
"<html>"
"<head>"
"<title>Arduino LED Control</title>"
"</head>"
"<body>"
"<form method=\"GET\">"
"<label for=\"led2\">Ukljuci LED 2</label>";

const char* html_end =
"</form>"
"</body>"
"</html>";

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 137, 177);
EthernetServer server(80); // inicijalizira na portu 80 (http)

void setup() {
    pinMode(LED2, OUTPUT);

    Serial.begin(9600);
    while (!Serial); // čekaj da se serial monitor otvori
```

```
Serial.println("\tEthernet webserver");
Ethernet.begin(mac, ip);

// ispitivanje je li ethernet shield priključen
if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    Serial.println("Ethernet shield nije pronađen.");
    while (true); // čekaj zauvijek, jer bez ethernet shielda nema
potrebe nastavljati kod
}
if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet kabel nije priključen.");
}

// pokreni server
server.begin();
Serial.print("Server je pokrenut na ");
Serial.println(Ethernet.localIP());
}

void loop() {
    // čekaj dok se jedan klijent ne spoji
    EthernetClient client = server.available();

    if (client) {
        String get_request = ""; // dio HTTP zahtjeva s GET zahtjevom
        (prva linija zahtjeva)
        Serial.println("New client connected:");

        // pročitaj dio HTTP zahtjeva koji sadrži GET zahtjev (prva
        linija),
        // spremi u get_request i ispiši na serial monitoru
        while (client.connected() && client.available()) {
            char c = client.read();
            get_request += c;
            Serial.print(c);
            if (c == '\n') break; // završi nakon prvog '\n' (kraj
        linije)
        }

        // pročitaj i ispiši do kraja
        while (client.connected() && client.available())
            Serial.print((char)client.read());

        client.print(http_response_header);
        client.print(html_start);
        bool led2_state = processCheckBox(client, get_request, "led2"); //
checkbox s imenom "led2"
        client.print(html_end);
        client.flush(); // pričekaj dok se sve nije poslalo

        // zatvori vezu
        client.stop();
    }
}
```



```
        Serial.println("client disconnected\n");

        digitalWrite(LED2, led2_state);
    }
}

// univerzalna funkcija za rukovanje checkboxa u GET zahtjevu
bool processCheckBox(EthernetClient client, String get_request, String name)
{
    // provjeri sadrži li GET zahtjev ime checkboxa s vrijednosti "on"
    bool checked = get_request.indexOf(name + "=on") > -1;

    // ispiši checkbox
    client.print("<input type=\"checkbox\" name=\"");
    client.print(name);
    client.print("\");
    if (checked) client.print(" checked"); // ako je checkbox označen,
    dodaj checked atribut
    client.print(" onclick=\"submit();\">");

    return checked;
}
```

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

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

#define LED2 2
#define POT A0

// http response header kojim će web server odgovoriti klijentu
// ovo će kompajler spojiti u jedan string, ovako je pisano čisto zbog
čitljivosti
const char* http_response_header =
"HTTP/1.1 200 OK\r\n"           // HTTP verzija 1.1, zahtjev je bio uspješan
"Content-Type: text/html\r\n"   // vraćam HTML
"Connection: close\r\n"        // nakon što dobiš podatke, zatvori vezu
"\r\n";                         // nakon prazne linije slijedi content

// početni, srednji i krajni dio web stranice
// između njih se nalazi sadržaj
// ovdje je korištenje "\r\n" opcionalno jer HTML nije briga za whitespace
(razmaci, znak za novu liniju)
const char* html_start =
"<!DOCTYPE html>"
"<html>"
"<head>"
"<title>Arduino Web</title>"
"</head>"
```

```
"<body>";

const char* html_mid =
"<form method=\"POST\">"
"<label for=\"led2\">Ukljuci LED 2</label>";

const char* html_end =
"</form>"
"</body>"
"</html>";

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 137, 177);
EthernetServer server(80); // inicijalizira na portu 80 (http)

void setup() {
    pinMode(LED2, OUTPUT);
    pinMode(POT, INPUT);

    Serial.begin(9600);
    while (!Serial); // čekaj da se serial monitor otvori
    Serial.println("\tEthernet webserver");
    Ethernet.begin(mac, ip);

    // ispitivanje je li ethernet shield priključen
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
        Serial.println("Ethernet shield nije pronađen.");
        while (true); // čekaj zauvijek, jer bez ethernet shielda nema
potrebe nastavljati kod
    }
    if (Ethernet.linkStatus() == LinkOFF) {
        Serial.println("Ethernet kabel nije priključen.");
    }

    // pokreni server
    server.begin();
    Serial.print("Server je pokrenut na ");
    Serial.println(Ethernet.localIP());
}

void loop() {
    // čekaj dok se jedan klijent ne spoji
    EthernetClient client = server.available();

    if (client) {
        String http_request = ""; // spremnik za HTTP zahtjev
        Serial.println("New client connected:");

        // pročitaj HTTP zahtjev, spremi u http_request i ispiši na serial
monitoru
        while (client.connected() && client.available()) {
            char c = client.read();
```

```
        http_request += c;
        Serial.print(c);
    }

    client.print(http_response_header);
    client.print(html_start);

    // ispis vrijednosti potencijometra
    client.print("<p>Vrijednost potencijometra: ");
    client.print(analogRead(POT));
    client.print("</p>");

    client.print(html_mid);
    bool led2_state = processCheckBox(client, http_request, "led2");
    // checkbox s imenom "led2"
    client.print(html_end);
    client.flush(); // pričekaj dok se sve nije poslalo

    // zatvori vezu
    client.stop();
    Serial.println("client disconnected\n");

    digitalWrite(LED2, led2_state);
}

// univerzalna funkcija za procesiranje checkboxa u GET zahtjevu
bool processCheckBox(EthernetClient client, String http_request, String name)
{
    // indeks od kojeg počinje sadržaj POST-a
    int POST_start = http_request.indexOf("\r\n\r\n") + 4;

    // provjeri sadrži li POST zahtjev ime checkboxa s vrijednosti "on"
    bool checked = http_request.indexOf(name + "=on", POST_start) > -1;

    // ispiši checkbox
    client.print("<input type=\"checkbox\" name=\"");
    client.print(name);
    client.print("\");
    if (checked) client.print(" checked"); // ako je checkbox označen,
    dodaj checked atribut
    client.print(" onclick=\"submit();\">");

    return checked;
}
```

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>

//#define SERVO_PIN 7

// http response header kojim će web server odgovoriti klijentu
// ovo će kompajler spojiti u jedan string, ovako je pisano čisto zbog
čitljivosti
const char* http_response_header =
"HTTP/1.1 200 OK\r\n"           // HTTP verzija 1.1, zahtijev je bio uspješan
"Content-Type: text/html\r\n"   // vraćam HTML
"Connection: close\r\n"        // nakon što dobiš podatke, zatvori vezu
"\r\n";                         // nakon prazne linije slijedi content

// HTML kod web stranice+
// ovdje je korištenje "\r\n" opcionalno jer HTML nije briga za whitespace
(razmaci, znak za novu liniju)
const char* html =
"<!DOCTYPE html>"
"<html>"
"<head>"
"<title>Arduino Web</title>"
"</head>"
"<body>"
"<p>Odaberi poziciju za servo:<br>"
"<a href=\"./?servo=on\">110</a><br>"
"<a href=\"./?servo=off\">10</a>"
"</p>"
"</body>"
"</html>";

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 137, 177);
EthernetServer server(80); // inicijalizira na portu 80 (http)
//Servo servo;

void setup() {
    pinMode(2, OUTPUT);

    Serial.begin(9600);
    while (!Serial); // čekaj da se serial monitor otvori
    Serial.println("\tEthernet webserver");
    Ethernet.begin(mac, ip);

    // ispitivanje je li ethernet shield priključen
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
        Serial.println("Ethernet shield nije pronađen.");
    }
}
```

```
        while (true); // čekaj zauvijek, jer bez ethernet shielda nema
potrebe nastavljati kod
    }
    if (Ethernet.linkStatus() == LinkOFF) {
        Serial.println("Ethernet kabel nije priključen.");
    }
    // pokreni server
    server.begin();
    Serial.print("Server je pokrenut na ");
    Serial.println(Ethernet.localIP());
    //servo.attach(SERVO_PIN);
}

void loop() {
    // čekaj dok se jedan klijent ne spoji
    EthernetClient client = server.available();

    if (client) {
        String get_request = ""; // spremnik za HTTP zahtjev
        Serial.println("New client connected:");

        // pročitaj dio HTTP zahtjeva koji sadrži GET (prva linija),
        // spremi u get_request i ispiši na serial monitoru
        while (client.connected() && client.available()) {
            char c = client.read();
            get_request += c;
            Serial.print(c);
            if (c == '\n') break; // završi nakon prvog '\n' (kraj
linije)
        }

        // pročitaj i ispiši do kraja
        while (client.connected() && client.available())
            Serial.print((char)client.read());

        client.print(http_response_header);
        client.print(html);
        client.flush(); // pričekaj dok se sve nije poslalo

        // zatvori vezu
        client.stop();
        Serial.println("client disconnected\n");

        if (get_request.indexOf("servo=on") > -1)
            //servo.write(110);
            digitalWrite(2, HIGH);

        if (get_request.indexOf("servo=off") > -1)
            //servo.write(10);
            digitalWrite(2, LOW);
    }
}
```


LITERATURA:

<https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/basic-web-server/>

<https://startingelectronics.org/tutorials/arduino/connecting-arduino-to-internet/>

<https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/web-server-LED-control/>