#include<SPI.h>

#include<Ethernet.h>

// http responseheader kojim će web server odgovoriti klijentu

// ovo će kompajler spojiti u jedan string, ovako je pisano čisto zbog čitljivosti

constchar\* 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)

constchar\* html\_start =

"<!DOCTYPE html>"

"<html>"

"<body>";

constchar\* html\_end =

"</body>"

"</html>";

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

IPAddressip(192, 168, 137, 177);

EthernetServer server(80); // inicijalizira na portu 80 (http)

voidsetup() {

Serial.begin(9600);

while (!Serial); // čekaj da se serial monitor otvori

Serial.println("\tEthernetwebserver");

Ethernet.begin(mac, ip);

// ispitivanje je li ethernetshield priključen

if (Ethernet.hardwareStatus() == EthernetNoHardware) {

Serial.println("Ethernet shield nije pronađen.");

while (true); // čekaj zauvijek, jer bez ethernetshielda 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());

}

voidloop() {

// čekaj dok se jedan klijent ne spoji

EthernetClientclient = server.available();

if (client) {

Serial.println("New clientconnected:");

// 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("clientdisconnected\n");

}

}