

# Arduino – Ethernet i primer relejne zaštite + DHT22

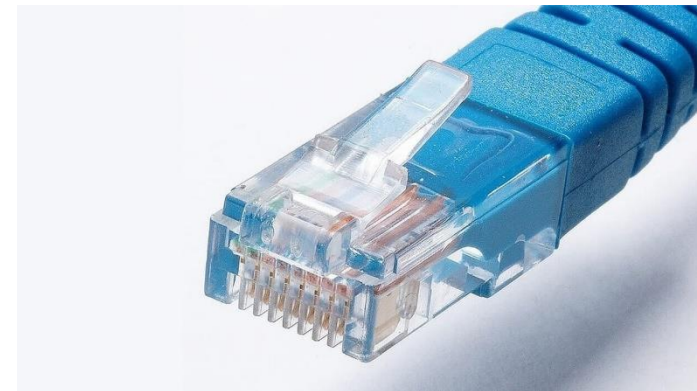
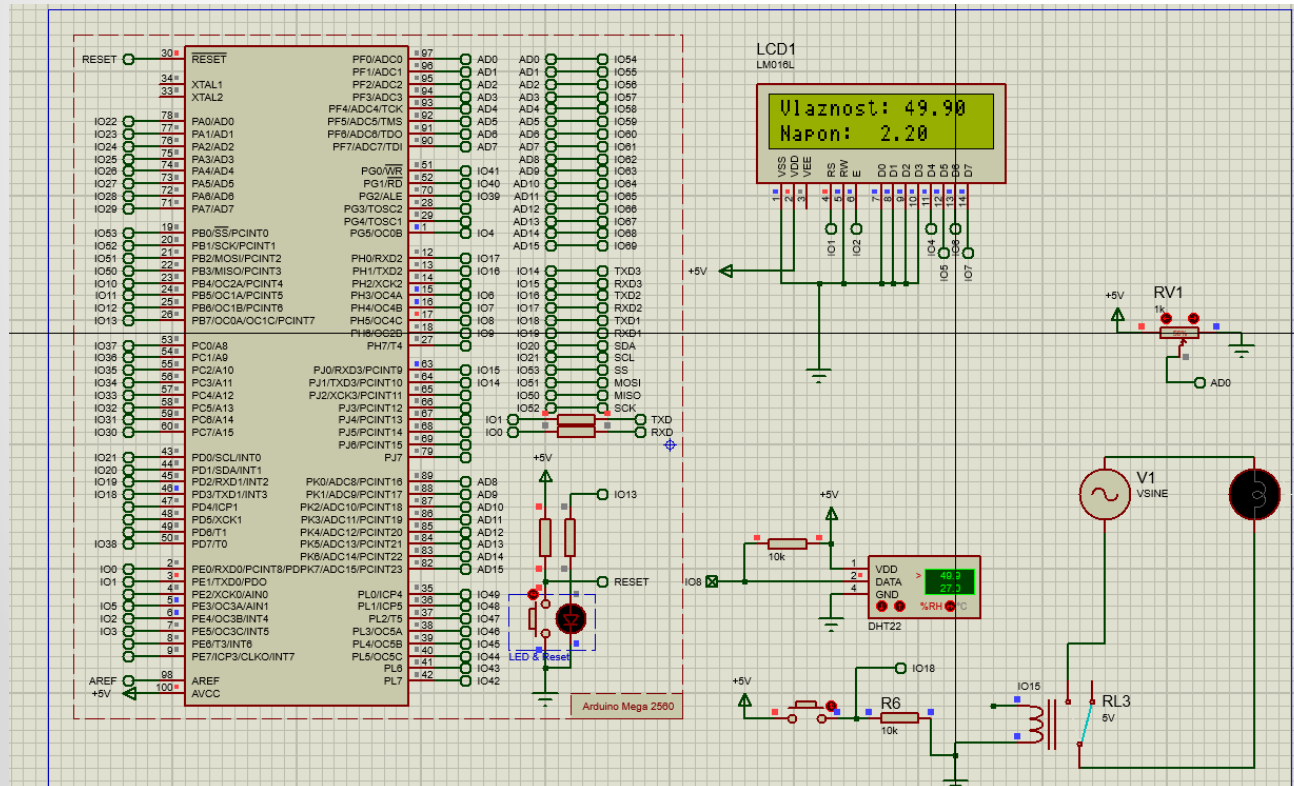


*Internet stvari 2023. - VII termin*

Nenad Petrović

Univerzitet u Nišu, Elektronski fakultet

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



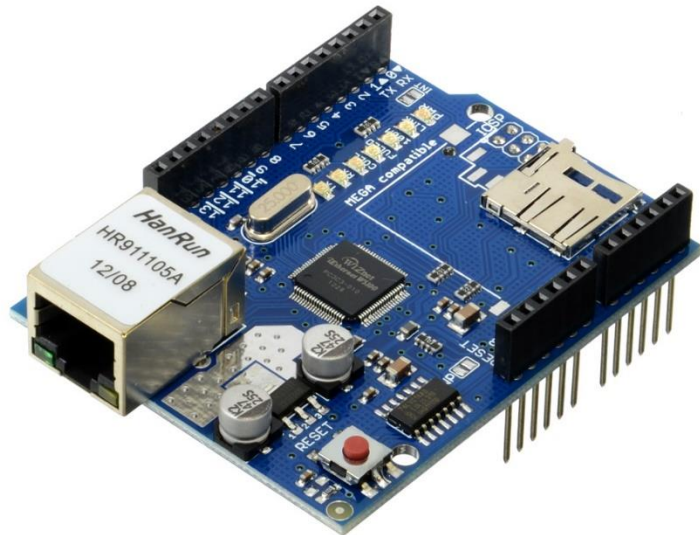
# Arduino i Ethernet Shield

## *Internet stvari 2023. - VII termin (1 čas)*

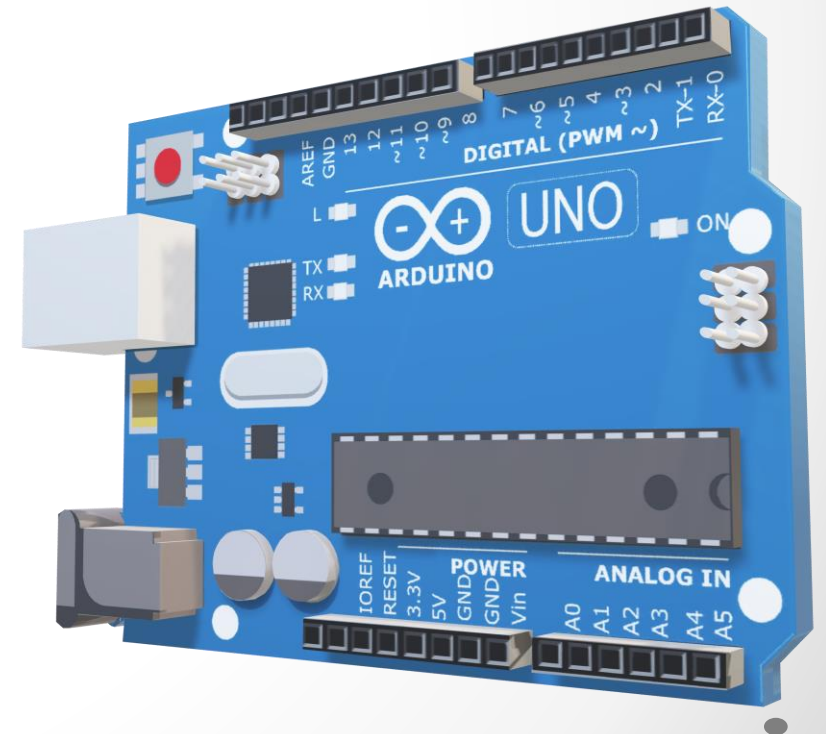
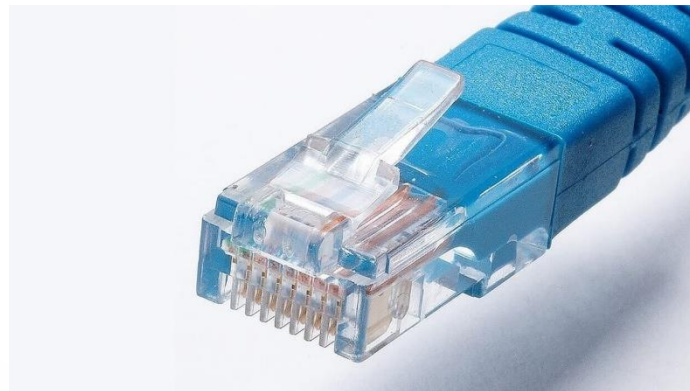
Nenad Petrović

Univerzitet u Nišu, Elektronski fakultet

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



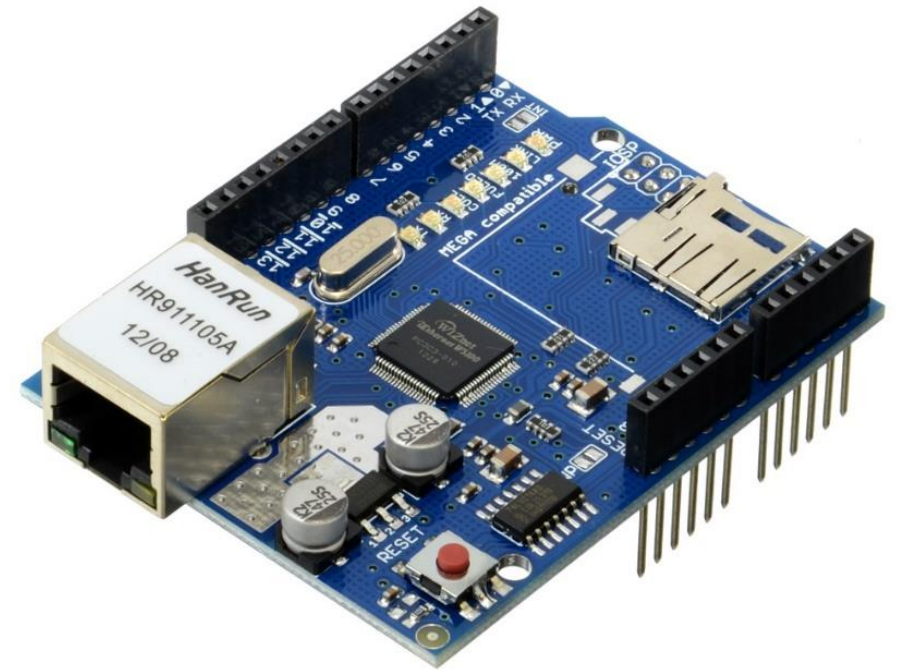
...





# Uvod

- Arduino Ethernet Shield omogućava prilično jednostavno povezivanje Arduina na lokalnu mrežu ili internet
- Zasnovan na Wiznet W5100 ili W5500 čipu
  - 10/100 – 10Mbps ili 100Mbps podržava
- Ovaj čip omogućava IP stek za TCP i UDP protokole
- Podržava do 8 istovremenih konekcija
- Za implementaciju programa za Arduino ploče se koristi Ethernet.h biblioteka
- Ethernet Shield ima standardni RJ-45 konektor, sa integrisanim linijskim transformatorom i napajanjem
- Postoji i (micro) SD slot za memorijske kartice
  - Služi za čuvanje fajlova koje pruža server
  - Kompatibilan sa Uno i Mega revizijama (Ethernet biblioteka)
  - Za pristup SD čitaču, koristi se SD.h biblioteka

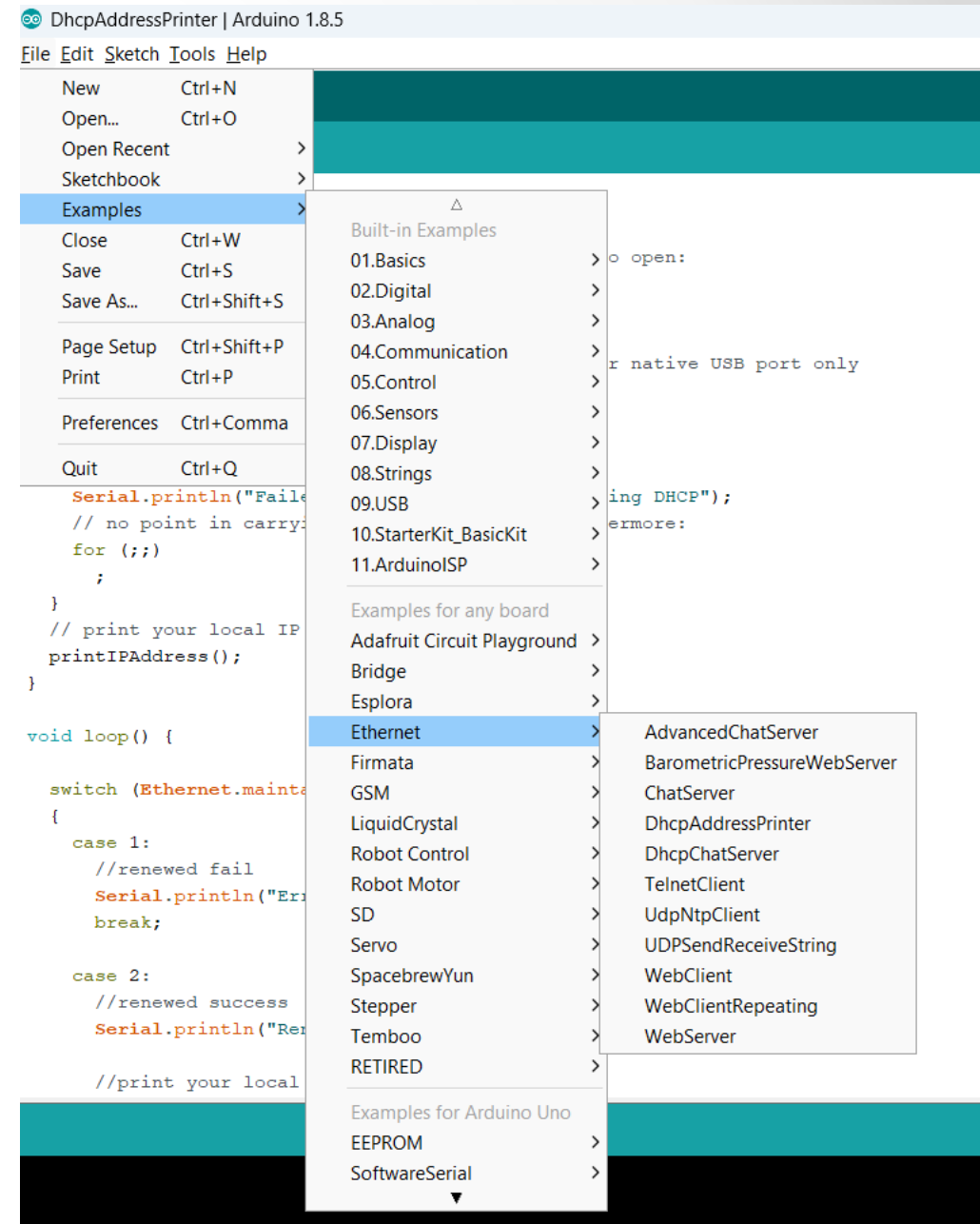


# Napomene

- Reset taster resetuje i Arduino i Shield
- Ne dolazi sa Power over Ethernet (PoE) modulom
  - Dodaje se zasebno
- Arduino komunicira sa W5500 i SD preko SPI magistrale (ICSP header)
  - Digitalni pinovi 10, 11, 12 i 13 na Uno
  - 50, 51 i 52 kod Mega
- Na obe ploče, pin 10 se koristi za selekciju W5500 i pin 4 za SD karticu
  - Ne mogu se koristiti za generalni I/O
- Kod Mega, hardverski Slave Select (SS) pin 53 se ne koristi da se selektuje niti W5500 niti SD, ali mora da bude u izlaznom režimu, inače SPI interfejs neće raditi.

# Napomene

- Postoji podrška za DHCP
  - U suprotnom, mora da se podesi IP ručno
- Arduino IDE nudi korisne gotove primere za rad sa Ethernetom
  - DhcpAddressPrinter koristimo kao probni program da saznamo dinamički dodeljenu IP adresu



# Ključni objekti i metode • Ethernet.begin()

## • IPAddress

- Predstavlja IP adresu Shield-a
- Parametri
  - uint8\_t first\_octet – prvi bajt adrese
  - uint8\_t second\_octet – drugi bajt adrese
  - uint8\_t third\_octet – treći bajt adrese
  - uint8\_t fourth\_octet – četvrti bajt adrese
- Primer inicijalizacije
  - **IPAddress ip(191,11,1,1);**

## • EthernetServer

- kreira server koji osluškuje konekcije na datom portu
- Parametar
  - Port na kome osluškuje, najviše 80
- Primer inicijalizacije
  - **EthernetServer server(80);**
  - Unutar setup-a
    - **server.begin();**

## • Ethernet.begin()

- Poziva se kao deo setup-a
- Parametri
  - mac
    - Štampana na samom Shield-u
    - Niz 6 bajtova koji predstavlja mac adresu
      - **byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDA, 0x02 };**
  - ip
    - IPAddress objekat, niz od 4 bajta
    - Default: zadnji oktet 1
  - dns
    - IP adresa DNS servera, 4 bajta
    - Default: zadnji oktet 1
  - gateway
    - IP adresa gateway-a, 4 bajta
  - subnet
    - Maska mreže, podrazumevano 255.255.255.0
- Varijante poziva
  - **Ethernet.begin(mac, ip);**
  - **Ethernet.begin(mac, ip, dns, gateway, subnet);**

# Ključni objekti i metode - nastavak

## • EthernetClient

- Kreira klijenta koji se može koristiti za konekciju na odgovarajuću IP adresu i port
  - **client.connect(server, port)**
- Primer slanja HTTP GET zahteva od strane klijenta

```
EthernetClient client;
if (client.connect(server, 80)) {
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET /search?q=arduino HTTP/1.1");
  client.println("Host: www.google.com");
  client.println("Connection: close");
  client.println();
}
```

- Primer slanja odgovora u vidu HTML strane od servera ka klijentu
  - Pokupljamo referencu klijenta, ako postoji njegov zahtev na serveru
  - Ako je klijent konektovan
    - Ako je dostupan klijent
      - Štampamo header
      - Štampamo stranicu

```
EthernetClient client = server.available();

while (client.connected()) {
  if (client.available()) {
    char c = client.read();
    // Kada dodjemo do novog reda, HTTP zahtev se
    završava, pa možemo da odogovorimo
    if (c == '\n' && currentLineIsBlank) {
      // HTTP header kao deo odgovora klijentu
      client.println("HTTP/1.1 200 OK");
      client.println("Content-Type: text/html");
      client.println();
      //Stranica
      client.println("<html><body>DOBRODOSLI!</body></html>");

      } break;
    }
  }
```

# Primer 7-1

- Realizovati serverski program koji se izvršava na Arduino sa Ethernet Shield-om i preko HTTP protokola osluškuje zahtev na portu 80, pri čemu daje različit sadržaj, zavisno da li je taster na pinu 2 pritisnut ili ne

```
//Uključujemo neophodne biblioteke za Ethernet
#include <SPI.h>
#include <Ethernet.h>

// Obično na shieldu piše MAC adresa, pročitati je
// IP adresa zavisi od lokalne mreže
byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDA, 0x02 };
IPAddress ip(191,11,1,1); //<<< NASA IP ADRESA!!!

// Inicijalizacija Ethernet server biblioteke
// Postavljamo IP adresu i port koji želimo da koristimo
// (port 80 je podrazumevani za HTTP):
EthernetServer server(80);

int buttonPress = 1;

void setup()
{
    //Povezan taster na pinu2
    pinMode(2, INPUT);

    //Zapoceti Ethernet konekciju:
    Ethernet.begin(mac, ip);
    //Pokrenuti server
    server.begin();
}
```

```
void loop()
{
    buttonPress = digitalRead(2);
    // osluškivanje zahteva klijenata
    EthernetClient client = server.available();
    if (client) {
        // HTTP zahtev koji se završava praznom linijom
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                // Kada dodjemo do novog reda, HTTP zahtev se završava, pa možemo da odgovorimo
                if (c == '\n' && currentLineIsBlank) {
                    // HTTP header kao deo odgovora klijentu
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-Type: text/html");
                    client.println();

                    //Različita stranica se generise zavisno da li je taster na pinu 2 pritisnut ili ne
                    if (buttonPress == 1) {
                        client.println("<html><body>SVETLO!</body></html>");
                    }
                    else if (buttonPress == 0){
                        client.println("<html><body>MRAK!</body></html>");
                    }

                    break;
                }
                if (c == '\n') {
                    // započinjemo novi red
                    currentLineIsBlank = true;
                }
                else if (c != '\r') {
                    // imamo karakter u trenutnoj liniji
                    currentLineIsBlank = false;
                }
            }
        }
        // dajemo vreme browseru klijenta da primi podatke
        delay(1);
        // zatvaramo konekciju
        client.stop();
    }
}
```



# Primer 7-2

- Realizovati klijentski program koji se izvršava na Arduinu sa Ethernet Shield-om koji šalje HTTP GET zahtev nekom serveru na internetu

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
char server[] = "www.google.com";    // name address for Google (using DNS)

// Statički IP ako DHCP ne radi
IPAddress ip(192, 168, 0, 177);

// Inicijalizacija klijenta
// port 80 je podrazumevani
EthernetClient client;

void setup() {
  // Čekamo da se port otvori:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  // započnemo konekciju - DHCP:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // Statički IP umesto DHCP
    Ethernet.begin(mac, ip);
  }
  //dajemo mu vremena da se inicijalizuje:
  delay(1000);
  Serial.println("connecting...");

  // ako se uspostavi veza:
  if (client.connect(server, 80)) {
    Serial.println("connected");
    // Make a HTTP request:
    client.println("GET /search?q=arduino HTTP/1.1");
    client.println("Host: www.google.com");
    client.println("Connection: close");
    client.println();
  } 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:
  if (client.available()) {
    char c = client.read();
    Serial.print(c);
  }

  // if the server's disconnected, stop the client:
  if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();

    // do nothing forevermore:
    while (true);
  }
}
```

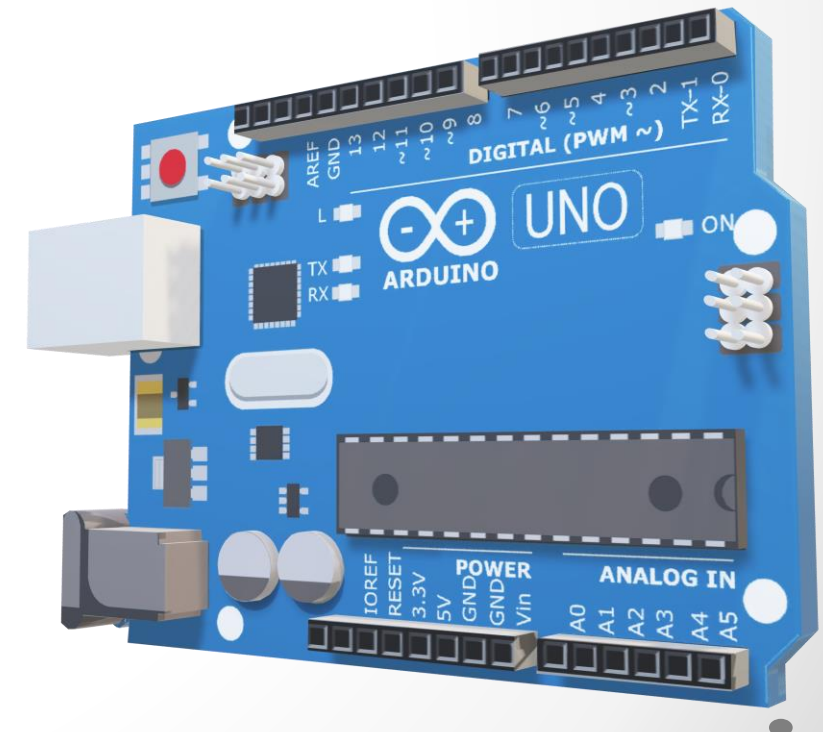
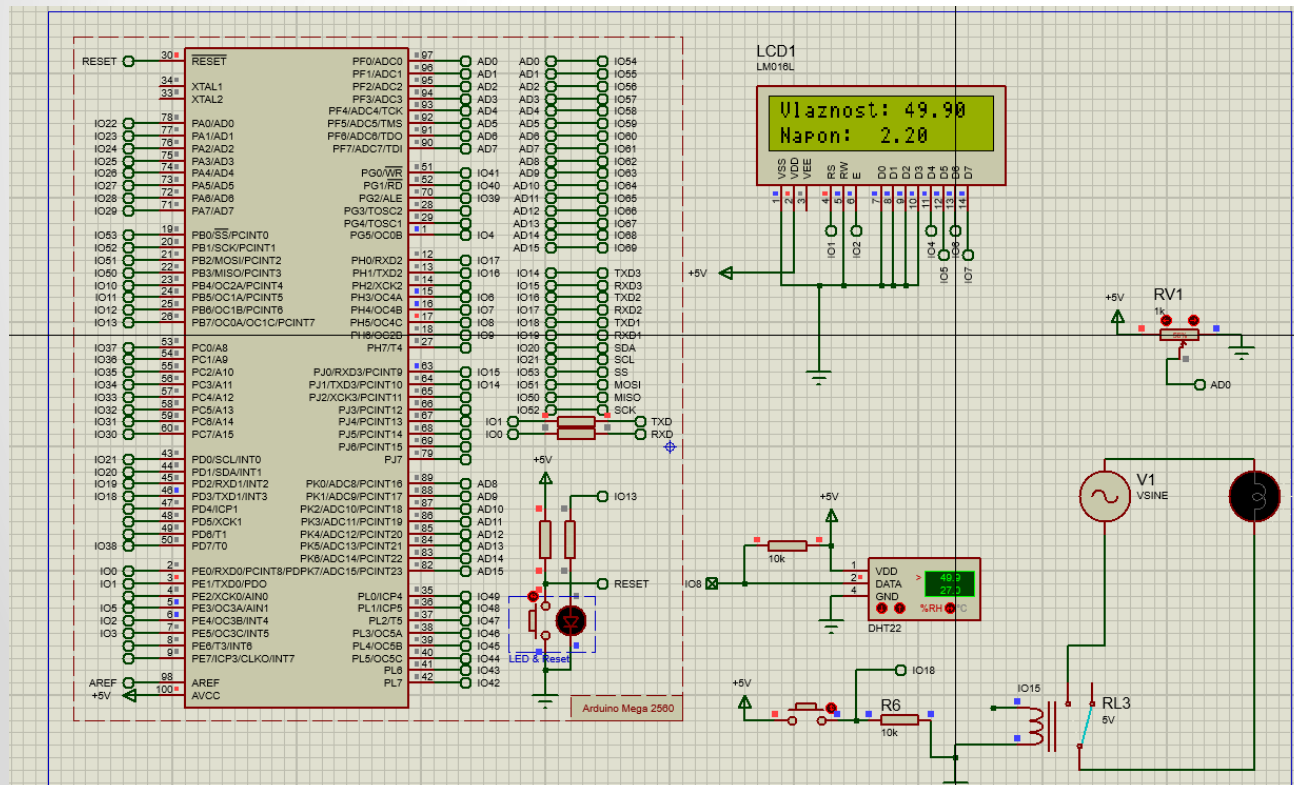
# Arduino – primer relejne zaštite + DHT22

## Internet stvari 2023. - VII termin (2 čas)

Nenad Petrović

Univerzitet u Nišu, Elektronski fakultet

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



# Primer 7-1

- Implementirati program za Arduino Mega koji služi za zaštitu uređaja (lampe) povezanog na izvor sinusnog naizmeničnih napona od vlage i većih varijacija napona u mreži.
- Za merenje relativne vlažnosti se koristi DHT22 senzor, a izmerene vrednosti ispisuju na LCD displeju. Svake 1.5 sekunde se meri relativna vlažnost. Ukoliko relativna vlažnost 50%, ugasi potrošača.
- Sa druge strane, na jednom od pinova Arduina je povezan potencijometar, koji simulira merni uređaj napona u mreži i njegove promene. Ukoliko je izmereni napon pao za 1V, isključiti uređaj preko releja. Promene napona se mere svake 3 sekunde.
- Konačno, prekidom aktiviranim pritiskom tastera omogućiti vraćanje sistema u početno stanje (uređaj upaljen).

# Rešenje 7-3

- Glavna petlja
  - checkHumid
  - Čita napon
  - Proverava da li je manji od 4V
    - Ako jeste, gasi lampu
- checkHumid
  - Proverava vlažnost
    - Aktivira relej ako je veća od 50% i gasi lampu
- resetSystem
  - Vraća u početno stanje

```
#include <LiquidCrystal.h>
#include "DHT.h"

#define DHTPIN 8
#define DHTTYPE DHT22

LiquidCrystal lcd(1,2,4,5,6,7);
DHT dht(DHTPIN, DHTTYPE, 4);

const int voltagePin = A0;
const int resetPin = 18;
const int relayPin = 15;
float humid = 0;
const float scale = 5.0/1023.0;
int voltageRead = 0;

void setup()
{
    pinMode(resetPin, INPUT);
    pinMode(relayPin, OUTPUT);
    pinMode(voltagePin, INPUT);
    digitalWrite(relayPin, HIGH);
    attachInterrupt(digitalPinToInterrupt(resetPin),
resetSystem, CHANGE);
    dht.begin();
    lcd.begin(16,2);
}
```

```
void loop()
{
    checkHumid();
    delay(1500);
    voltageRead = analogRead(voltagePin);
    float voltage = voltageRead * scale;
    if(voltage <= 4){ // Posto je potencijometar
povezan na napajanje od 5V, ukoliko je procitani
manji od 4V treba da se ugasi potrosac
        digitalWrite(relayPin, LOW);
    }
    lcd.setCursor(0,1);
    lcd.print("Napon: ");
    lcd.setCursor(8,1);
    lcd.print(voltage);
    delay(1500);
}
```

```
void resetSystem(){
    digitalWrite(relayPin, HIGH);
}

void checkHumid(){
    lcd.clear();
    humid = dht.readHumidity();
    if(isnan(humid)){
        lcd.clear();
        lcd.print("Neispravno!");
        return;
    }
    lcd.print("Vlaznost: ");
    lcd.setCursor(10, 0);
    lcd.print(humid);
    if(humid > 50){
        lcd.clear();
        lcd.print("Visoka vlaznost!");
        digitalWrite(relayPin, LOW);
    }
}
```