

# PAMETNA UTIČNICA

---

**Pavičić, Ian Karlo**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Karlovac University of Applied Sciences / Veleučilište u Karlovcu**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:128:440550>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-13**



**VELEUČILIŠTE U KARLOVCU**  
Karlovac University of Applied Sciences

*Repository / Repozitorij:*

[Repository of Karlovac University of Applied Sciences - Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VELEUČILIŠTE U KARLOVCU  
STROJARSKI ODJEL  
STRUČNI PRIJEDIPLOMSKI STUDIJ MEHATRONIKA

IAN KARLO PAVIČIĆ

# **PAMETNA UTIČNICA**

ZAVRŠNI RAD

KARLOVAC, 2024.

VELEUČILIŠTE U KARLOVCU  
STROJARSKI ODJEL  
STRUČNI PRIJEDIPLOMSKI STUDIJ MEHATRONIKA

IAN KARLO PAVIČIĆ

# **PAMETNA UTIČNICA**

ZAVRŠNI RAD

Mentor: Filip Žugčić, mag. ing. el.

KARLOVAC, 2024.

 <b>VELEUČILIŠTE U KARLOVCU</b> Karlovac University of Applied Sciences	Klasa: 602-07/___-01/___
<b>ZADATAK ZAVRŠNOG / DIPLOMSKOG RADA</b>	Datum:  <b>11.07.2024.</b>

* Ime i prezime	Ian Karlo Pavičić		
OIB / JMBG			
Adresa			
Tel. / Mob./e-mail			
Matični broj studenta			
JMBAG			
Studij (staviti znak <b>X</b> ispred odgovarajućeg studija)	<input checked="" type="checkbox"/> preddiplomski	<input type="checkbox"/> specijalistički diplomski	
Naziv studija	Stručni prijediplomski studij Mehatronika		
Godina upisa	2020.		
Datum podnošenja molbe	11.07.2024.		
Vlastoručni potpis studenta/studentice			

* Naslov teme na hrvatskom: Pametna utičnica	
* Naslov teme na engleskom: Smart plug	
Opis zadatka: U završnom radu bit će prikazana izrada pametne utičnice korištenjem Arduina i ESP32 mikroupravljača prema pravilniku Veleučilišta u Karlovcu.	
Mentor:	Predsjednik Ispitnog povjerenstva:

## IZJAVA O SAMOSTALNOSTI RADA

Izjavljujem da sam samostalno izradio ovaj rad.

Karlovac, 2024.

---

Ime i Prezime

## **SAŽETAK**

Tema ovog završnog rada je prikaz kompozicije i analiza pametnih utičnica. Objasnit će programski dio, konstrukciju, te analizirati ponudu i potražnju na tržištu pametnih utičnica. Fokusirat će se na svaku individualnu komponentu potrebnu za funkcionalni rad pametne utičnice, te njen daljnji razvitak u okruženju ostalih pametnih uređaja.

Ključne riječi: Pametna utičnica, programski dio, konstrukcija, tržište

## **SUMMARY**

The topic of this final paper is the presentation of the composition and analysis of smart plugs. It will explain the program and codes necessary for usage, the construction and analyze the supply and demand in the smart plug market. It will focus on each individual component necessary for the functional operation of the smart plug and its further development in the environment of the smart devices.

Key words: Smart plug, program and codes, construction, market

**POPIS SLIKA**

Slika 1: X10 timer [2].....	2
Slika 2: Modul lampe [4].....	3
Slika 3: Arduino UNO [8].....	5
Slika 4: Spoj komponenti [Autor].....	16
Slika 5: Nadžbukna kutija [Autor].....	18
Slika 6: Arduino ESP32 [Autor].....	19
Slika 7: Relej 5V [Autor].....	20
Slika 8: Malo napajanje [Autor].....	21
Slika 9: Upaljena utičnica [Autor].....	21
Slika 10: Ugašena utičnica [Autor].....	22
Slika 11: Sklop [Autor].....	23
Slika 12: Prednja strana kućišta [Autor].....	23
Slika 13: Stražnja strana kućišta [Autor].....	23



**POPIS TABLICA**

Tablica 1: Značenje prvog broja IPXX zaštite [12].....	17
Tablica 2: Značenje drugog broja IPXX zaštite [12].....	17

**POPIS OZNAKA**

Oznaka:	Opis:
<i>V</i>	Volt
<i>DC</i>	Istosmjerna struja
<i>AC</i>	Izmjenična struja

**SADRŽAJ**

<b>1. UVOD</b> .....	1
<b>2. POVIJEST PAMETNIH UTIČNICA</b> .....	2
2.1. Rani počeci (1980. – 1990.) .....	2
2.2. Razvoj interneta stvari (IoT) (2000.) .....	3
2.3. Uspon pametnih kućnih sustava (2010.) .....	3
2.4. Današnje stanje i budući razvoj .....	4
2.4.1. Prednosti .....	4
2.4.2. Nedostatci .....	4
<b>3. PROGRAMIRANJE</b> .....	5
3.1. Arduino .....	5
3.2. Objašnjenje programa po segmentima .....	6
3.3. Objašnjenje programa po funkcijama .....	11
3.3.1. Funkcija updateStatusColor .....	11
3.3.2. Funkcija handleRoot .....	11
3.3.3. Funkcija handleOn .....	12
3.3.4. Funkcija handleOff .....	12
3.3.5. Funkcija handleStatus .....	13
3.3.6. Funkcija handleNotFound .....	13
3.3.7. Funkcija loop .....	14
<b>4. MATERIJALI I KONSTRUKCIJA UTIČNICE</b> .....	16
4.1. Materijali .....	16
4.1.1. Arduino ESP32 .....	18
4.1.2. Releji .....	19
4.1.3. Malo napajanje (transformator) .....	20
4.2. Način rada .....	21
<b>ZAKLJUČAK</b> .....	24
<b>LITERATURA</b> .....	25

## 1. UVOD

Pametna utičnica je mali adapter koji se može priključiti u zidnu utičnicu na koju se može spojiti pomoću Wi-Fi mreže [1]. Koncept potječe još iz osamdesetih godina prošlog stoljeća, no tek značajniji napredak u upotrebi svakodnevnog života dobivaju u ovome stoljeću. Pametna utičnica predstavlja jedan od ključnih elemenata modernih pametnih kućnih sustava. Korisnicima nakon spajanja omogućuje daljinsko upravljanje bilo kojeg uređaja koji je priključen na adapter pametne utičnice [1]. Sve većom integracijom tehnologije u svakodnevni život, pametne utičnice postale su neizostavan dio suvremenih domova. One pružaju niz prednosti uključujući uštedu energije, povećanu sigurnost i praktičnost.

## 2. POVIJEST PAMETNIH UTIČNICA

Pametne utičnice su dio širokog trenda razvoja pametnih kućnih tehnologija, koji je započeo s razvojem interneta stvari (IoT). Njihova povijest može se pratiti kroz nekoliko ključnih faza.

### 2.1. Rani počeci (1980. – 1990.)

Početak 1980., X10 protokol postao je široko prihvaćeni standard za komunikaciju između elektroničkih uređaja unutar kućanstva. X10 uređaji koristili su električne instalacije za prijenos signala, omogućujući osnovne funkcije daljinskog upravljanja. Rane pametne utičnice imale su ograničene mogućnosti, uglavnom su se svodile na osnovne funkcije uključivanja i isključivanja uređaja putem daljinskog upravljača.



Slika 1. X10 timer [2]

Protokol je razvijen 1975. godine od strane Pico Electronics iz Škotske. U to vrijeme sustav se sastojao od 16-kanalne komandne konzole, modula lampe i modula uređaja. Ubrzo nakon toga pojavio se zidni sklopni modul i prvi X10 timer. [3]



Slika 2. Modul lampe [4]

## 2.2. Razvoj interneta stvari (IoT) (2000.)

Kevin Ashton 1999. godine prvi put je izrekao frazu Internet of things kako bi opisao sistem u kojem je internet povezan s fizičkim stvarima pomoću senzora. [5]

Razvojem novih komunikacijskih protokola kao što su Zigbee i Z-Wave, pametne utičnice počele su dobivati naprednije mogućnosti. IoT ili internet stvari je mreža povezanih fizičkih uređaja čija je svrha razmjenjivati podatke s ostalim uređajima koji imaju pristup internetu. U uređaje su ugrađene softverske tehnologije čiji je cilj povezivanje s ostalim uređajima unutar mreže bez ljudske pomoći. Pojavom Wi-Fi tehnologije, pametne utičnice su se počele povezivati s internetom, omogućujući daljinsko upravljanje putem mobilnih aplikacija i računalnih sustava.

## 2.3. Uspon pametnih kućnih sustava (2010.)

Pojavom pametnih mobitela dolazi do sve veće popularnosti pametnih utičnica zbog mogućnosti upravljanja putem mobilnih aplikacija. Korisnici su u mogućnosti nadzirati i upravljati svojim uređajima s bilo koje lokacije. Razvojem glasovnih asistenata kao što su Amazon-ova Alexa, Google-ov Assistant, te Apple-ova Siri, pametne utičnice su integrirane u širi spektar pametnih kućnih uređaja, omogućujući glasovno upravljanje uređaja.

## 2.4. Današnje stanje i budući razvoj

Danas pametne utičnice nude brojne napredne mogućnosti, uključujući praćenje potrošnje energije u stvarnom vremenu, automatsko uključivanje i isključivanje uređaja, zaštitu od prenapona i povezivanje s drugim pametnim kućnim sustavima. Gledajući na rastuće zabrinutosti oko sigurnosti i privatnosti, proizvođači pametnih utičnica počeli su ugrađivati napredne sigurnosne protokole i metode enkripcije kako bi zaštitili korisničke podatke i uređaje od mogućih napada. Pametne utičnice su evoluirale od jednostavnih uređaja za daljinsko upravljanje do naprednih komponenti modernih domova, pružajući veću sigurnost i kontrolu. Očekuje se da će daljnji razvoj tehnologije pridodati funkcionalnosti pametnih utičnica.

### 2.4.1. Prednosti

Pametne utičnice donose puno prednosti unutar kućanstva, čineći svakodnevni život sigurnijim, lakšim i energetski efikasnijim. Korisnicima omogućavaju praćenje potrošnje energije u stvarnom vremenu putem pametnih telefona što pomaže pri prepoznavanju uređaja s najvećom potrošnjom i omogućuju donošenje odluka oko njihovog korištenja. Korisnici mogu daljinski regulirati radno stanje utičnica putem aplikacija, smanjujući rizik od požara ili nekih drugih faktora koji mogu uzrokovati štetu. Bitno je zaštititi osjetljive elektroničke uređaje, te iz tog razloga puno pametnih utičnica dolazi sa zaštitom od prenapona.

### 2.4.2. Nedostatci

Unatoč tome što pametne utičnice donose mnoge prednosti imaju i brojne nedostatke koje bi trebalo uzeti u obzir prije njihove ugradnje u svoj dom. Pametne utičnice obično su skuplje od standardnih utičnica što uvelike povećava početni trošak pogotovo ako se planira ugradnja po cijelom kućanstvu. IoT uređaji među koje spadaju i pametne utičnice mogu biti podložne hakerskim napadima što može ugroziti sigurnost osobnih podataka i mreže. Također ovise o stabilnoj internetskoj vezi što može biti problem u manjim naseljima i ruralnim područjima. Unatoč svim ovim nedostacima, korisnici smatraju da prednosti ipak nadmašuju nedostatke.

### 3. PROGRAMIRANJE

Programiranje je pisanje uputa računalu što i kako učiniti, a izvodi se u nekom od programskih jezika. Programiranje je umjetnost i umijeće u stvaranju programa za računala. Stvaranje programa sadrži u sebi pojedine elemente dizajna, umjetnosti, znanosti, matematike kao i inženjeringa. [6]

#### 3.1. Arduino

Arduino je open-source platforma za kreiranje elektroničkih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje. Arduino platforma je skup elektroničkih i softverskih komponenti koje se mogu jednostavno povezivati u složenije cjeline s ciljem izrade zabavnih i poučnih elektroničkih sklopova [7]. Srce Arduina su mikrokontroleri. Mikrokontroler je malo računalo sadržano na jednom integriranom sklopu. Arduino okruženje najčešće koristi 8 bitne mikrokontrolere koje proizvodi tvrtka ATMEL. Najrasprostranjeniji model je ATMEGA328P koji se koristi na osnovnoj Arduino prototipnoj pločici.



Slika 3. Arduino UNO [8]



### 3.2. Objašnjenje programa po segmentima

```
#include <WiFi.h>
#include <WebServer.h>
#include <DNSServer.h>

const char *ssid = „Pametna_Uticnica“;
const char *password = „12345678“;

WebServer server(80); // Port 80 na kojem će web server slušati dolazne http zahtjeve
DNSServer dnsServer;

const int relejPin = 2; // Ugrađeni LED na ESP32 ploči
const byte DNS_PORT = 53; // Port 53 koji se koristi za DNS usluge. DNS prevodi
domene u IP adrese i obrnuto

// HTML web stranica za upravljanje LED-om
const char htmlPage[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <title>Pametna Uticnica</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body { font-family: Arial, sans-serif; text-align: center; margin-top: 50px; }
    .button { display: inline-block; padding: 15px 25px; font-size: 24px; cursor: pointer; text-
align: center; text-decoration: none; outline: none; color: #fff; border: none; border-radius:
15px; box-shadow: 0 9px #999; }
    .button-on { background-color: #4CAF50; }
    .button-on:hover { background-color: #45a049; }
    .button-on:active { background-color: #45a049; box-shadow: 0 5px #666; transform:
translateY(4px); }
    .button-off { background-color: #f44336; }
    .button-off:hover { background-color: #da190b; }
```

```
.button-off:active { background-color: #da190b; box-shadow: 0 5px #666; transform:
translateY(4px); }
.status { font-size: 24px; margin-top: 20px; }
footer { margin-top: 50px; font-size: 18px; color: #777; }
</style>
</head>
<body>
<h1>Pametna Uticnica</h1>
<button class="button button-on" onclick="toggleLED('on')">Upali</button>
<button class="button button-off" onclick="toggleLED('off')">Ugasi</button>
<p class="status" id="status">Status: Nepoznato</p>
<p id="datetime"></p>
<footer>Firmware verzija 1.0</footer>
<script>
function toggleLED(action) {
  fetch('/' + action)
    .then(response => response.text())
    .then(data => {
      document.getElementById('status').innerText = 'Status: ' + data;
      updateStatusColor(data);
    });
}

function updateStatusColor(status) {
  if (status.trim() === 'Upaljeno') {
    document.body.style.backgroundColor = '#d4edda';
  } else {
    document.body.style.backgroundColor = '#f8d7da';
  }
}

// Dohvati početni status
fetch('/status')
  .then(response => response.text())
```

```
.then(data => {
  document.getElementById('status').innerText = 'Status: ' + data;
  updateStatusColor(data);
});

// Ažuriraj datum i vrijeme svake sekunde
setInterval(() => {
  const now = new Date();
  document.getElementById('datetime').innerText = now.toLocaleString();
}, 1000);
</script>
</body>
</html>
)rawliteral“;

// Rukovatelj početne stranice
void handleRoot() {
  server.send(200, „text/html“, htmlPage);
}

// Rukovatelj za uključivanje utičnice
void handleOn() {
  digitalWrite(relejPin, HIGH);
  server.send(200, „text/plain“, „Upaljeno“);
}

// Rukovatelj za isključivanje utičnice
void handleOff() {
  digitalWrite(relejPin, LOW);
  server.send(200, „text/plain“, „Ugaseno“);
}

// Rukovatelj za statusni zahtjev
void handleStatus() {
```

```
if (digitalRead(relejPin) == HIGH) {
  server.send(200, „text/plain“, „Upaljeno“);
} else {
  server.send(200, „text/plain“, „Ugaseno“);
}
}

// Ovo će preusmjeriti nepoznate HTTP zahtjeve na glavnu stranicu kontrole
void handleNotFound() {
  server.sendHeader(„Location“, „/“);
  server.send(302, „text/plain“, „redirect na stranicu kontrole“);
}

void setup() {
  pinMode(relejPin, OUTPUT); // Postavljanje relej pina kao izlaz
  digitalWrite(relejPin, LOW); // Isključivanje relej pina na početku da početno stanje bude
  ugašeno

  Serial.begin(115200); // Pokretanje komunikacije na serial portu 115200 baud

  // Postavljanje Wi-Fi pristupne točke
  WiFi.softAP(ssid, password);
  Serial.println();
  Serial.print(„IP Adresa: „); // Ispis ip adrese na serial portu radi debuga
  Serial.println(WiFi.softAPIP()); // Ispis IP adrese pristupne točke

  // Postavljanje captive portala
  dnsServer.start(DNS_PORT, „*“, WiFi.softAPIP());

  // Server root
  server.on(„/“, handleRoot);

  // Rukovatelj za uključivanje utičnice
  server.on(„/on“, handleOn);
```

```
// Rukovatelj za isključivanje utičnice
server.on(„/off“, handleOff);

// Rukovatelj za status utičnice
server.on(„/status“, handleStatus);

// Svi nepoznati zahtjevi se preusmjeravaju na stranicu za kontrolu uticnice
server.onNotFound(handleNotFound);

server.begin(); // Pokretanje HTTP poslužitelja
Serial.println(„HTTP server pokrenut“); // Debug na serial port
}

void loop() {
  dnsServer.processNextRequest(); // Obrada DNS zahtjeva
  server.handleClient(); // Obrada HTTP klijenata
}
```

### 3.3. Objašnjenje programa po funkcijama

#### 3.3.1. Funkcija updateStatusColor

Učinkovita ali jednostavna funkcija za promjenu boje pozadine web stranice ovisno o statusu koji se prosljeđuje kao argument. Ova funkcija prima jedan status, koji se koristi za određivanje je li status „Upaljeno“ ili nešto drugo, te na temelju toga postavlja boju pozadine stranice na odgovarajuću vrijednost. Funkcija radi na sljedeći način:

- Prvo, poziva se metoda trim() koja uklanja sve nepotrebne razmake s početka i kraja stringa kako bi se izbjegle pogreške prilikom usporedbe.
- Drugo, funkcija provjerava je li status jednak stringu 'Upaljeno'. Ako je taj uvjet zadovoljen, boja pozadine web stranice postavlja se na #d4edda, što je svijetlo zelena nijansa. Ova boja obično simbolizira pozitivno stanje, pa se koristi kako bi se naznačilo da je sustav „upaljen“. U suprotnom, ako status nije 'Upaljeno', boja pozadine mijenja se u #f8d7da, što je svijetlo crvena nijansa. Ova boja često je povezana s pogreškama ili upozorenjima, i koristi se kako bi se prikazalo da je sustav „ugašen“ ili da postoji neki problem.

#### 3.3.2. Funkcija handleRoot

Ključna funkcija u postavljanju jednostavnog web poslužitelja na uređajima poput ESP8266 ili ESP32. Ova funkcija koristi se za rukovanje glavnom (root) stranicom web poslužitelja i odgovorna je za slanje odgovora kada klijent (npr. web preglednik) zatraži pristup toj stranici. Funkcija radi na sljedeći način:

- handleRoot koristi funkcionalnost servera definiranog unutar programa kako bi poslala odgovor klijentu.
- Unutar funkcije, metoda server.send() poziva se s tri argumenta:
  - Prvi argument (200) predstavlja HTTP statusni kod koji označava uspješan zahtjev. Kod 200 OK standardna je oznaka za uspješno izvršavanje zahtjeva.
  - Drugi argument („text/html“) označava vrstu sadržaja koji se šalje klijentu. U ovom slučaju, to je HTML sadržaj, koji web preglednici interpretiraju kako bi prikazali web stranicu.

- Treći argument (htmlPage) sadrži stvarni HTML sadržaj koji će biti poslan klijentu. Ovaj sadržaj obično je definiran kao string unutar programa i predstavlja strukturu i izgled web stranice.

### 3.3.3. Funkcija handleOn

Ključna funkcija je za upravljanje i uključivanje releja putem web poslužitelja na mikroprocesorskim platformama poput ESP8266 ili ESP32. Ova funkcija omogućuje korisniku da daljinski uključi relej i dobije povratnu informaciju o stanju uređaja. Funkcija radi na sljedeći način:

- Uključivanje releja: Prvi redak unutar funkcije, `digitalWrite(relejPin, HIGH);`, postavlja logičku razinu na određenom digitalnom pinu na visoki napon (HIGH). Taj pin povezan je s relejom, a kada je na visokom naponu, relej se aktivira i zatvara krug, što uzrokuje uključivanje povezanog uređaja (npr. svjetla, motora ili drugog električnog opterećenja).
- Slanje odgovora klijentu: Nakon uključivanja releja, funkcija koristi metodu `server.send()` kako bi poslala odgovor klijentu koji je inicirao zahtjev.
- Prvi argument (200) označava HTTP statusni kod, što znači da je zahtjev uspješno obrađen.
- Drugi argument („text/plain“) definira vrstu sadržaja koji se šalje, u ovom slučaju običan tekst (plain text).
- Treći argument („Upaljeno“) je tekstualna poruka koja informira klijenta da je relej uspješno uključen.

### 3.3.4. Funkcija handleOff

Funkcija `handleOff` za razliku od funkcije `handleOn` služi za isključivanje releja putem web poslužitelja.

### 3.3.5. Funkcija handleStatus

Omogućavanje korisniku da provjeri trenutno stanje releja putem web poslužitelja na mikroprocesorskoj platformi poput ESP8266 ili ESP32. Ova funkcija čita stanje određenog digitalnog pina na kojem je spojen relej i vraća odgovarajuću tekstualnu poruku koja označava je li relej uključen ili isključen. Funkcija radi na sljedeći način:

- Provjera stanja releja: Funkcija koristi `digitalRead (relejPin)` kako bi pročitala trenutnu logičku razinu (visoki ili niski napon) na pinu kojem je pridružen relej.
- Ako je vrijednost koju `digitalRead` vraća `HIGH` (visoki napon), to znači da je relej uključen. U tom slučaju, funkcija šalje odgovor „Upaljeno“.
- Ako je vrijednost `LOW` (niski napon), relej je isključen, te funkcija šalje odgovor „Ugašeno“.
- Slanje odgovora klijentu:
  - Funkcija koristi metodu `server.send()` kako bi poslala tekstualni odgovor klijentu:
    - Prvi argument (`200`) označava uspješan HTTP statusni kod.
    - Drugi argument („`text/plain`“) specificira da se šalje običan tekst.
    - Treći argument ovisi o stanju releja: „Upaljeno“ ako je relej uključen, ili „Ugašeno“ ako je isključen.

### 3.3.6. Funkcija handleNotFound

Rukovanje zahtjevima koji nisu povezani s definiranim URL rutama na web poslužitelju postavljenom na mikroprocesorskoj platformi poput ESP8266 ili ESP32. Kada korisnik pokuša pristupiti stranici koja ne postoji ili koja nije definirana u programu, ova funkcija automatski preusmjerava korisnika na početnu stranicu (root stranicu) poslužitelja. Funkcija radi na sljedeći način:

- Postavljanje zaglavlja za preusmjeravanje:
  - Funkcija koristi `server.sendHeader(„Location“, „/“)`; kako bi postavila HTTP zaglavlje `Location`. Ovo zaglavlje specificira URL na koji će korisnik biti preusmjeren. U ovom slučaju, to je korijenska („/“) stranica poslužitelja.



- Slanje HTTP odgovora:
  - Nakon postavljanja zaglavlja za preusmjeravanje, funkcija koristi `server.send(302, „text/plain“, „redirect na stranicu kontrole“)`; za slanje HTTP odgovora s kodom 302.
  - HTTP statusni kod 302 Found označava da je resurs privremeno premješten na drugu lokaciju, a web preglednik automatski slijedi preusmjeravanje na URL specificiran u zaglavlju Location.
  - Drugi argument („text/plain“) označava da je vrsta sadržaja običan tekst.
  - Treći argument je poruka „redirect na stranicu kontrole“, koja obavještava klijenta o preusmjeravanju. Iako većina preglednika automatski slijedi preusmjeravanje i ne prikazuje ovu poruku, ona može biti korisna u nekim slučajevima (npr. za logiranje ili debugging).

### 3.3.7. Funkcija loop

Funkcija `loop()` srce je svakog Arduino programa i izvršava se kontinuirano dok je uređaj uključen. U ovom konkretnom slučaju, funkcija `loop()` upravlja obradom DNS i HTTP zahtjeva na mikroprocesorskoj platformi poput ESP8266 ili ESP32, omogućujući uređaju da reagira na mrežne zahtjeve u realnom vremenu. Funkcija `loop()` radi na sljedeći način:

Obrada DNS zahtjeva:

- Prva linija unutar funkcije, `dnsServer.processNextRequest()`, zadužena je za obradu dolaznih DNS zahtjeva. DNS (Domain Name System) služi za prevođenje imena domena u IP adrese. U kontekstu ESP8266 ili ESP32, DNS poslužitelj se često koristi za implementaciju lokalnog DNS-a, koji može omogućiti korisnicima pristup uređaju putem jednostavnog URL-a umjesto IP adrese.
- Metoda `processNextRequest()` pregledava i obrađuje svaki DNS zahtjev koji stigne na uređaj, omogućujući korisnicima da koriste prilagođena imena domena za pristup uređaju.

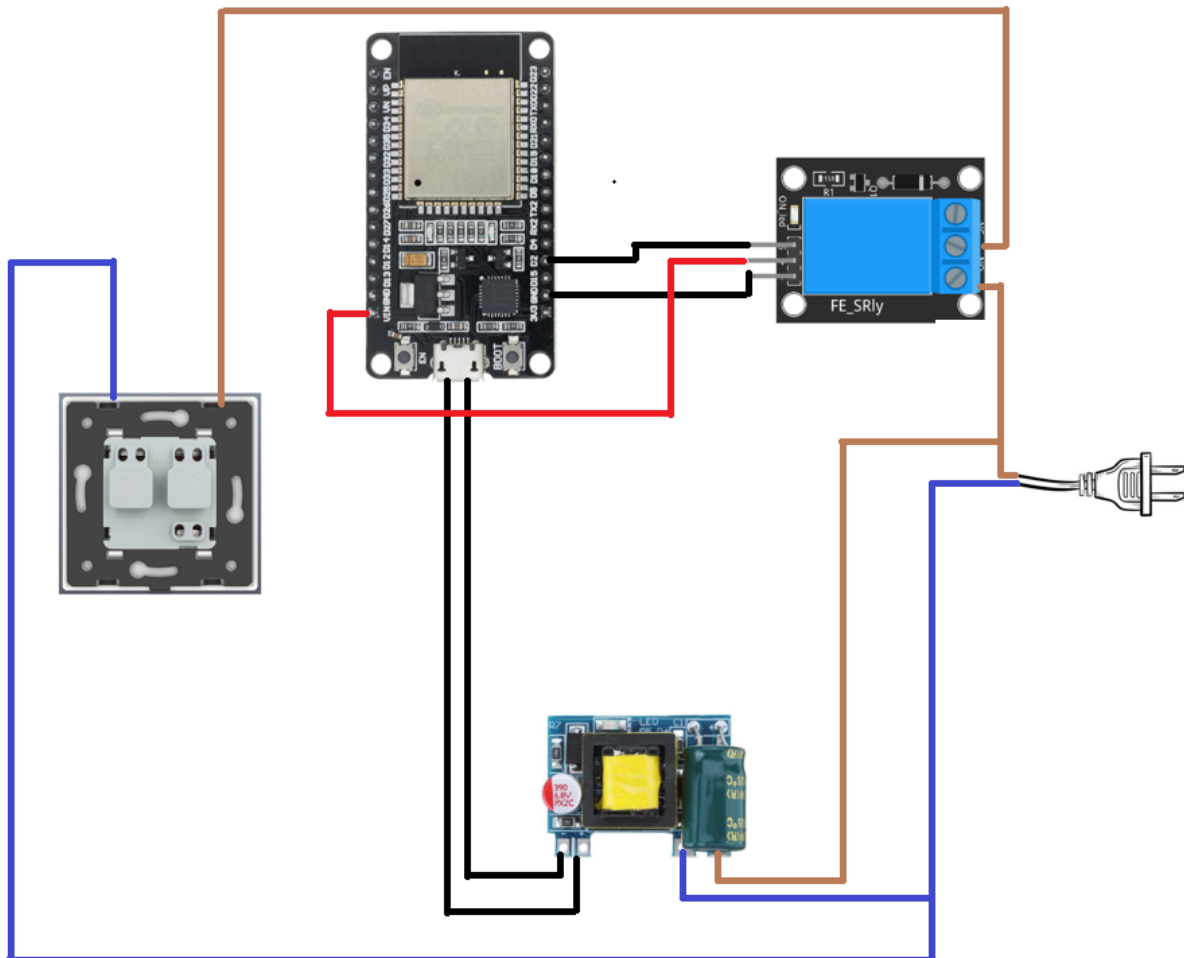
Obrada HTTP klijenata:

- Druga linija, `server.handleClient()`, zadužena je za obradu HTTP zahtjeva koje klijenti (npr. web preglednici) šalju poslužitelju na uređaju.

- Metoda `handleClient()` provjerava ima li novih zahtjeva od klijenata i, ako ih ima, obrađuje ih. To može uključivati slanje odgovora na zahtjeve za web stranicama, izvođenje akcija poput uključivanja ili isključivanja releja, preusmjerenje korisnika na druge stranice, i slično.

## 4. MATERIJALI I KONSTRUKCIJA UTIČNICE

Prikaz spojenih komponenti unutar kućišta vidljiva je na shemi ispod. Stvarni prikaz spoja biti će vidljiv u nastavku ove cjeline nakon objašnjenja pojedinih komponenti.



Slika 4. Spoj komponenti [Autor]

### 4.1. Materijali

Površinski dio sastoji se od nadžbukne kutije koja sa jedne strane sadrži utičnicu dok se sa druge strane nalazi utikač. Kutija na sebi sadrži oznaku IP55 što označava vrstu mehaničke zaštite. Oznake stupnja zaštite označene su oblikom IP XX [12].

Prva brojka označava stupanj zaštite od dodira i stranih tijela. Ta se zaštita ostvaruje onemogućavanjem ili ograničavanjem ulaska predmeta koji osoba drži u kućište. Stupanj zaštite se označava brojevima od 0 do 6.

Tablica 1: Značenje prvog broja IPXX zaštite [12]

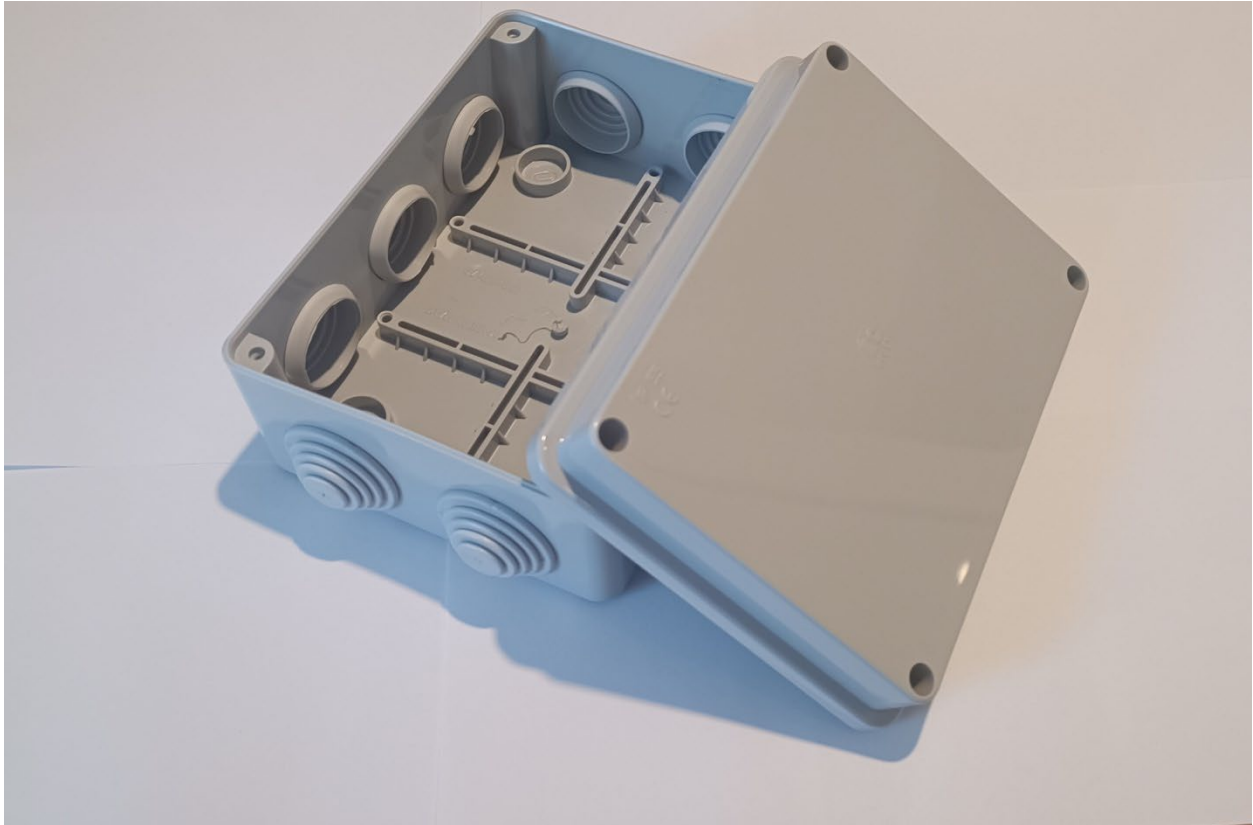
Prva brojka zaštite	
0.	Bez zaštite
1.	Zaštićeno od krutih stranih tijela promjera 50mm i većih
2.	Zaštićeno od krutih stranih tijela promjera 12,5mm i većih
3.	Zaštićeno od krutih stranih tijela promjera 2,5mm i većih
4.	Zaštićeno od krutih stranih tijela promjera 1,0mm i većih
5.	Zaštita od prašine
6.	Ne propušta prašinu

Druga brojka označava vrstu zaštite kućišta koja jamči sigurnost u slučaju prodiranja vode u kućište. Stupanj zaštite se označava brojevima od 0 do 9, a njihova značenja su prikazana u tablici 2.

Tablica 2: Značenje drugog broja IPXX zaštite [12]

Drugi broj zaštite	
0.	Bez Zaštite
1.	Zaštićeno od vode koja kapa
2.	Zaštićeno od vode koja kapa ako je kućište nagnuto do 15°
3.	Zaštićeno od vode koja pršti
4.	Zaštićeno od vode koja prska
5.	Zaštićeno od mlaza vode
6.	Zaštićeno od snažnog mlaza vode
7.	Zaštićeno od djelovanja pri povremenom uranjanju u vodu
8.	Zaštićeno od djelovanja pri trajnom uranjanju u vodu
9.	Voda pri čišćenju visokim tlakom/mlazom pare

Unutrašnjost se sastoji od Arduina ESP32 zaduženog za upravljanje, malog napajanja koje vrši pretvorbu napona te releja namijenjenog za upravljanjem protoka struje ovisno o namjeri korisnika koji to kontrolira pomoću pametnog telefona.



Slika 5. Nadzbukna kutija [Autor]

#### 4.1.1. Arduino ESP32

Arduino Nano ESP32 je prva Arduino pločica koja se temelji na mikrokontroleru ESP32 tvrtke Espressif, modulu NORA-W106 tvrtke U-blox. Omogućena joj je podrška za MicroPython & Arduino Cloud, te je vrlo svestrana razvojna ploča. ESP32 razvojna ploča je prikladna za napredne projekte koji trebaju veće mogućnosti obrade i povezanost kao što su; pametne kućne aplikacije, industrijska automatizacija i IoT uređaji. Jednostavnost Arduino ploče omogućit će lako korištenje u obrazovne svrhe. ESP32 nudi veliki broj digitalnih i analognih ulaza/izlaza, uključujući PWM, SPI, I2C, i UART, što omogućuje povezivanje raznih senzora, aktuatora i drugih uređaja. Podržava različite načine rada s niskom potrošnjom energije, što ga čini pogodnim i za baterijski napajane projekte.



Slika 6. Arduino ESP32 [Autor]

#### 4.1.2. Relej

Relejni modul je elektromehanički modul prekidača koji se može koristiti za upravljanje visokonaponskim uređajima s niskonaponskim mikrokontrolerom, kao što je Arduino. Obično ima tri pina: VCC (napajanje), GND (uzemljenje) i IN (ulazni kontrolni signal).



Slika 7. Relej 5V [Autor]

### 4.1.3. Malo napajanje (transformator)

Električni transformator je uređaj koji bez pokretnih dijelova međuinduktivno povezuje dva električna kruga izmjenične struje i pretvara izmjeničnu struju određenog napona u izmjeničnu struju višeg ili nižeg napona. Glavni dijelovi transformatora su magnetska jezgra i barem dva namota (poznata kao primar i sekundar) s izolacijom (osim kod autotransformatora). U jednofaznom transformatoru primar i sekundar obično sadrže jedan namot, dok trofazni transformator ima tri namota. Primarni i sekundarni namoti najčešće su postavljeni jedan iznad drugoga ili jedan pored drugoga kako bi se osigurala optimalna međuinduktivna veza. Električna energija prenosi se elektromagnetskom indukcijom s primara na sekundar, bez promjene frekvencije.



Slika 8. Malo napajanje [Autor]

Glavna funkcija malog napajanja ili transformatora u programiranju Arduina za pametnu utičnicu je osigurati stabilan i odgovarajući napon i struju za rad Arduina i povezanih komponenti. Ovo napajanje omogućuje:

1. **Napajanje Arduina:** Osiguravanje potrebnog napona (obično 5V ili 3.3V) za stabilan rad mikrokontrolera.
2. **Napajanje senzora i modula:** Osiguravanje napajanja za različite senzore, releje, Wi-Fi module ili druge komponente koje se koriste u pametnoj utičnici.

3. **Stabilnost i sigurnost:** Pružanje stabilnog izvora napajanja kako bi se spriječili problemi s nestabilnim naponom koji mogu uzrokovati nepredvidivo ponašanje ili oštećenje uređaja.
4. **Izolacija i sigurnost:** U nekim slučajevima, transformator može pružiti izolaciju između visokog napona iz mreže i niskog napona koji koristi Arduino, što povećava sigurnost cijelog sustava.

Sve ove funkcije omogućuju pouzdan rad pametne utičnice, kontrolirane putem Arduina, osiguravajući da svi elektronički dijelovi rade optimalno.

## 4.2. Način rada

Sklop radi na način da ESP32 služio kao mozak operacije. Programiranjem se na njega stavlja kod u kojemu se nalazi i web stranica na kojoj su dva gumba, on i off. Pritiskom na jedan od dva gumba u pozadini se pali ili gasi pin 2, ovisno o funkciji koju odabere korisnik. Pin 2 je spojen na signalni pin od releja na kojemu se nalaze još 2 pina, + i -, ta dva pina predstavljaju napajanje za relej. Signalni pin na releju se pali nakon što do njega dođe 3.3V, u ovome slučaju pali ga pritisak na gumb „on“.



Slika 9. Upaljena utičnica [Autor]

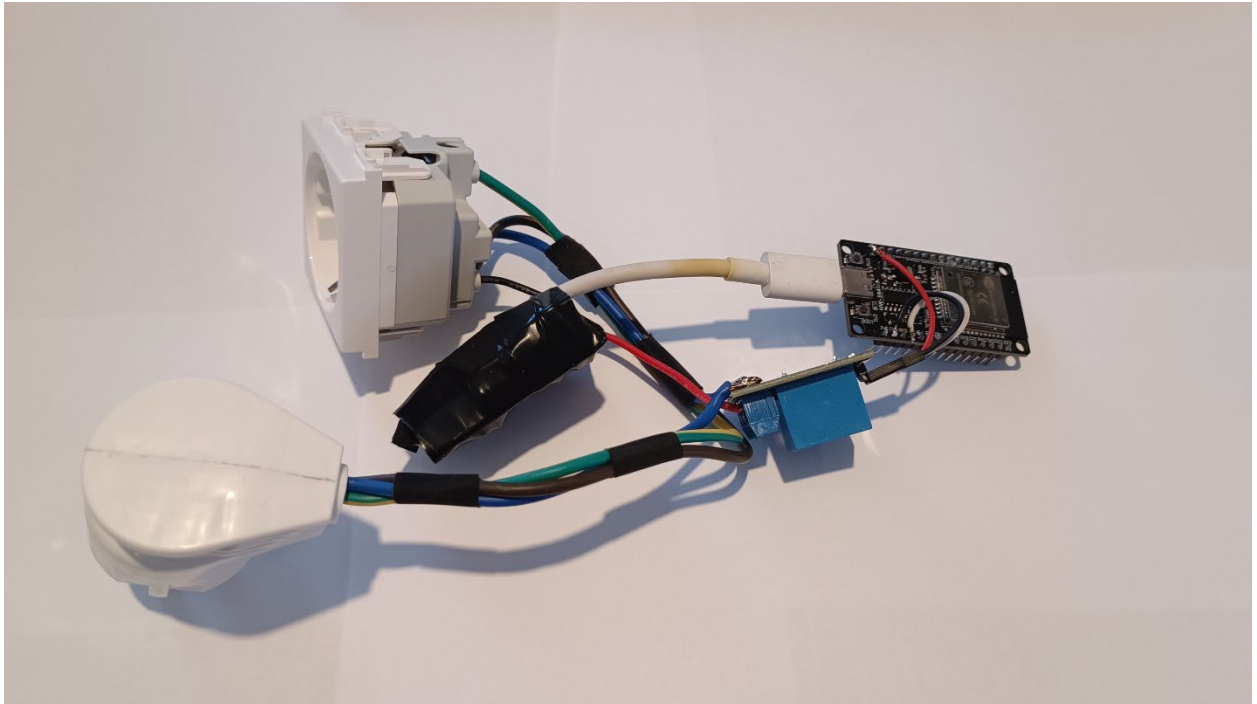


Pritiskom na gumb „off“ pin 2 daje 0V pa se relej gasi.



Slika 10. Ugašena utičnica [Autor]

Koristimo napajanje od mobitela kao glavni dovod struje pošto ESP zahtjeva napajanje preko USB-a koji koristi 5V. Pretvorba se dešava na ESP pločici koja tih 5V pretvara u 3.3V. Malo napajanje sa transformatorom u sredini koristimo jedino da 220V AC pretvorimo u 5V DC.

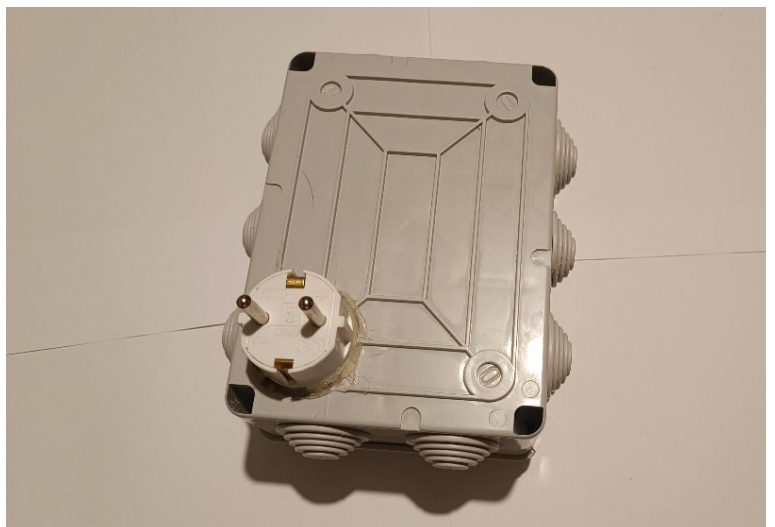


Slika 11. Sklop [Autor]

Kućište je bilo potrebno ručno izrezati. Prednju stranu u obliku pravokutnika radi utičnice, te stražnju stranu u obliku kruga radi utikača.



Slika 12. Prednja strana kućišta [Autor]



Slika 13. Stražnja strana kućišta [Autor]

## ZAKLJUČAK

Konstrukcija i programiranje pametne utičnice za završni rad bio je edukativan zadatak jednako koliko je bio izazovan. U procesu izrade završnog rada bilo je od velike pomoći prijašnje znanje stečeno tijekom školovanja, te mogućnost laganog nadograđivanja tog znanja radi puno dostupnih resursa. Korištene su osnove poznavanja elektronike, programiranja i dostupnih materijala i komponenti. Pored toga, integracija dodatnih sigurnosnih funkcija mogla bi značajno povećati pouzdanost uređaja u svakodnevnoj upotrebi. Pametne utičnice unatoč svojim nedostacima na kraju se ipak pokazuju kao korisna stvar u kućanstvima kao i cjelokupni IoT. Potencijal za povezivanje pametne utičnice s mobilnim aplikacijama omogućava korisnicima dodatnu kontrolu i uvid u potrošnju energije, što dodatno doprinosi efikasnosti. Budući rad na ovom projektu mogao bi uključivati i povezivanje s pametnim asistentima poput Google Assistant-a ili Amazon Alexe, čime bi se omogućila još jednostavnija upotreba. Ovaj projekt ne samo da predstavlja važan korak u razumijevanju IoT tehnologija, već i doprinosi sve većem trendu automatizacije u modernim domaćinstvima.

## LITERATURA

[1] DIY Smart Home Solutions, "What Is a Smart Plug? How They Work and How to Use Them", URL:

<https://www.diysmarthomesolutions.com/what-is-a-smart-plug-how-they-work-and-how-to-use-them/> , pristupljeno 13. srpnja 2024.

[2] X10.com, "Home", URL:

<https://www.x10.com/>, pristupljeno 13. srpnja 2024.

[3]Wikipedia,"X10(industrystandard)",URL:

[https://en.wikipedia.org/wiki/X10\\_\(industry\\_standard\)](https://en.wikipedia.org/wiki/X10_(industry_standard)) , pristupljeno 13. srpnja 2024.

[4] Powerhouse, "X10 LM12 Lamp Module", URL:

<https://www.powerhouse.eu/en/plug-in-modules/1-x10-lm12-lamp-module.html>,

pristupljeno 14. srpnja 2024.

[5] Duplico, "Što je IoT ili Internet of Things", URL:

<https://duplico.io/sto-je-iot-ili-internet-of-things/>, pristupljeno 15. srpnja 2024.

[6]Wikipedia,"Računalnoprogramiranje",URL:

[https://hr.wikipedia.org/wiki/Ra%C4%8Dunalno\\_programiranje](https://hr.wikipedia.org/wiki/Ra%C4%8Dunalno_programiranje), pristupljeno 15. srpnja 2024.

[7] E-Elektro Blog, "Što je Arduino?", URL:

<https://e-elektro.blogspot.com/2014/06/sto-je-arduino.html>, pristupljeno 16. srpnja 2024.

[8] Wikipedia, "Arduino Uno",URL:

[https://en.wikipedia.org/wiki/Arduino\\_Uno](https://en.wikipedia.org/wiki/Arduino_Uno), pristupljeno 16. srpnja 2024.

[9] Circuito.io, "Home", URL:

<https://www.circuito.io/>, pristupljeno 19. srpnja 2024.

[10] Arduino, "Arduino Guide", URL:

<https://www.arduino.cc/en/Guide>, pristupljeno 20. srpnja 2024.

[11] Instructables, "Home", URL:

<https://www.instructables.com/> , pristupljeno 20. srpnja 2024.

[12] Regulator, „Zanimljivosti“, URL:

<https://regulator.hr/zanimljivosti/ip-stupnjevi-zastite/>, pristupljeno 30. kolovoza 2024.