

Razvoj ugradbenih sustava

## Projekt

Ak. god. 2022./2023.

# Arduino Meteo Stanica

*Daniel Peruško*

Datum predaje: 25.08.2023.

## Sadržaj

Sadržaj.....	2
Slike .....	3
Tablice .....	4
Dnevnik promjena dokumentacije.....	5
Pojmovnik.....	6
1 Opis projektnog zadatka.....	7
2 Opis sustava .....	8
2.1 Motivacija projekta.....	8
2.2 Resursi.....	8
2.3 Minimalni zahtjevi .....	9
3 Specifikacija zahtjeva .....	10
3.1 Pretpostavke .....	12
3.2 Funkcijski zahtjevi .....	12
3.2.1 Opća funkcionalnost.....	12
3.2.2 Funkcionalnosti načina rada.....	12
3.2.3 Funkcionalnosti nadzornog načina rada .....	13
3.2.4 Zahtjevi napajanja .....	13
3.3 Mehanički zahtjevi.....	13
3.3.1 Dimenzije i kućište .....	13
3.4 Ostali zahtjevi.....	13
3.5 Cijena.....	13
3.6 Buduće značajke.....	14
4 Detaljnije specifikacije funkcije .....	15
4.1 Implementacija i korisničko sučelje.....	15
4.1.1 Oblikovanje sklopovlja .....	15
4.1.2 Sučelje sklopovlja.....	16
4.1.3 Oblikovanje programske potpore.....	16
4.1.4 Implementacija .....	17
5 Korištene tehnologije i alati.....	21
6 Zaključak i budući rad.....	22
7 Reference.....	23

## Slike

Slika 1. Dijagram slučajeva uporabe.....	10
Slika 2. Dijagram aktivnosti. ....	11
Slika 3. Dijagram slijeda.....	12
Slika 4. Oblikovanje sklopa. ....	15
Slika 5. Web aplikacija. ....	17
Slika 6. Prvi dio programskog koda. ....	18
Slika 7. Drugi dio programskog koda.....	19
Slika 8. Ispis konzole.....	20

## Tablice

Tablica 1. Dnevnik promijena dokumentacije .....	5
Tablica 2. Opća funkcionalnost. ....	12
Tablica 3. Funkcionalnosti načina rada. ....	13
Tablica 4. Funkcionalnosti nadzornog načina rada. ....	13
Tablica 5. Zahtjevi napajanja. ....	13
Tablica 6. Cijena.....	13
Tablica 7. Buduće značajke.....	14

## Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak.	Daniel Peruško	20.06.2023.
0.2	Nabava opreme.	Daniel Peruško	21.06.2023.
0.3	Pisanje programskog koda. Testiranje.	Daniel Peruško	25.06.2023.
0.4	Pisanje seminara i izrada dijagrama.	Daniel Peruško	26.6.2023.

*Tablica 1. Dnevnik promijena dokumentacije*

## Pojmovnik

**Arduino** - Arduino je otvorena platforma za razvoj i prototipiziranje elektronskih projekata. Korisnicima omogućava jednostavno programiranje mikrokontrolera i interakciju s različitim sensorima, motorima i drugim elektronskim komponentama.

**C++** - Programski jezik.

**JavaScript** - Programski jezik.

**DTH 11** - jednostavan digitalni senzor temperature i vlage.

**Fotootpornik** - pasivni elektronički komponent koji mijenja svoj električni otpor ovisno o količini svjetlosti koja ga pogađa.

**Lokalna mreža** - komunikacijska mreža koja povezuje različite uređaje unutar relativno malog fizičkog područja.

**Web aplikacija** - softverska aplikacija koja se izvršava putem web preglednika korisnika. Umjesto da se instalira na lokalnom uređaju, web aplikacija se nalazi na udaljenom serveru i korisnici je pristupaju putem internetskog preglednika.

**Bluetooth** - bežična tehnologija za kratkodometnu komunikaciju između različitih uređaja.

# 1 Opis projektnog zadatka

Cilj ovog projekta je razviti malu meteorološku stanicu koja će prikupljati podatke o temperaturi, vlažnosti zraka i jačini sunčeve svjetlosti. Ti podaci će biti prikazani putem web aplikacije, omogućavajući korisnicima praćenje trenutanih vremenskih uvjeta na odabranoj lokaciji.

## 2 Opis sustava

Ovaj projekt predstavlja vremensku stanicu koja koristi ESP8266 mikrokontroler za prikupljanje i prikaz trenutnih meteoroloških podataka. Koristeći DHT22 senzor, mjeri temperaturu i vlažnost zraka, dok fotoresistor očitava svjetlinu okoline. Podaci se koriste za praćenje uvjeta u okolini.

Kroz web sučelje, dostupno lokalno putem preglednika, korisnici mogu pratiti očitane vrijednosti. Na stranici se prikazuju aktualne temperature, vlažnost i svjetlina, čime korisnicima omogućuje trenutni uvid u uvjete okoline. Web sučelje također sadrži opciju brisanja povijesti očitavanja, što olakšava praćenje promjena tijekom vremena.

Osim toga, projekt omogućava interakciju putem Arduino IDE konzole. Unosom naredbi poput "TEMP", "HUM" ili "LIGHT", korisnici mogu dobiti trenutne vrijednosti relevantnih parametara direktno u konzoli.

Ova vremenska stanica može biti korisna u kućnim okruženjima, vrtovima, školama i drugim mjestima gdje je važno pratiti temperaturu, vlažnost i svjetlo. Projekt omogućava jednostavno praćenje i analizu uvjeta okoline, što može imati primjene u područjima kao što su poljoprivreda, unutarnje uzgoj biljaka, kućna automatizacija i slično.

### 2.1 Motivacija projekta

Cilj nam je pružiti korisnicima praktičan način praćenja vremenskih uvjeta uz pomoć malog uređaja i jednostavne web aplikacije.

### 2.2 Resursi

- ESP8266 mikrokontroler: ESP8266 je temelj projekta i koristi se za čitanje senzorskih podataka, uspostavu WiFi veze i vođenje web servera.
- DHT11 senzor: DHT11 je senzor za mjerenje temperature i vlažnosti zraka.
- Fotoresistor: Fotoresistor se koristi za mjerenje svjetla.
- Arduino IDE: Arduino razvojno okruženje za pisanje i učitavanje koda na ESP8266.
- WiFi mreža: Mora postojati dostupna WiFi mreža kako bi se ESP8266 mogao povezati i pristupiti internetu.
- Preglednik: Web preglednik za pristup web sučelju vaše vremenske stanice.
- HTML i CSS znanje
- Fizikalne komponente: ESP8266, DHT11 senzor, fotoresistor, žice i ostale komponente za povezivanje.
- Napajanje: ESP8266 treba napajanje, bilo putem USB kabela ili baterija.
- Računalo: Potrebno je računalo za pisanje koda, preuzimanje biblioteka i učitavanje koda na ESP8266

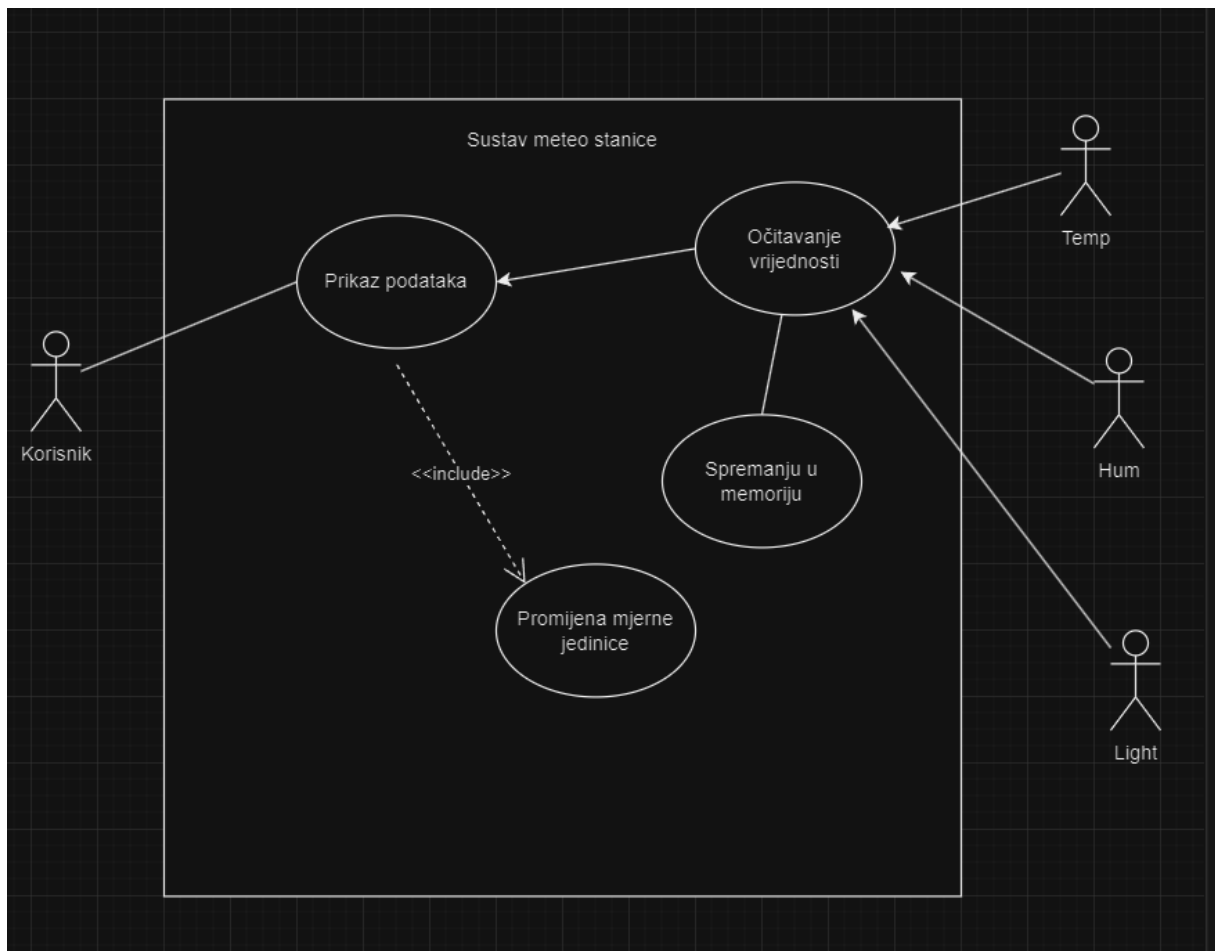


## **2.3 Minimalni zahtjevi**

Projekt treba osigurati točnost prikupljenih podataka, stabilnost rada, jednostavno korisničko sučelje i pouzdanu komunikaciju između senzora, mikrokontrolera i web aplikacije.

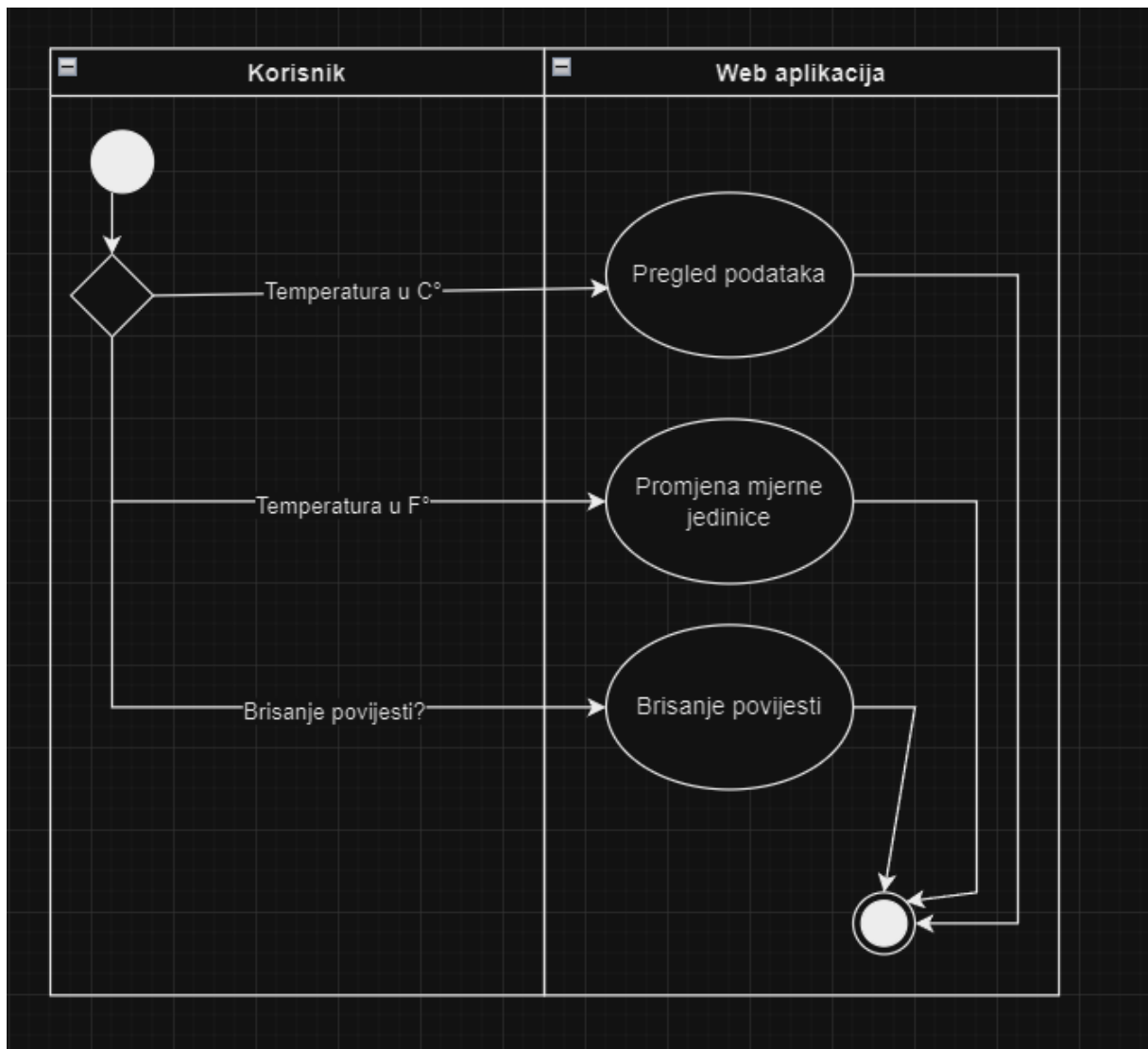
### 3 Specifikacija zahtjeva

Na slici 1. prikazan je dijagram slučajeva uporabe. Dijagram se sastoji od četiri aktora. Tri senzora koji šalju informacije mikrokontroleru. Zatim se ti podatci ispisuju u web aplikaciji gdje korisnik vidi te podatke i zapisuju u kratkotrajnu memoriju. Web aplikaciji također dodjeljujemo funkciju promijene mjerne jedinice.



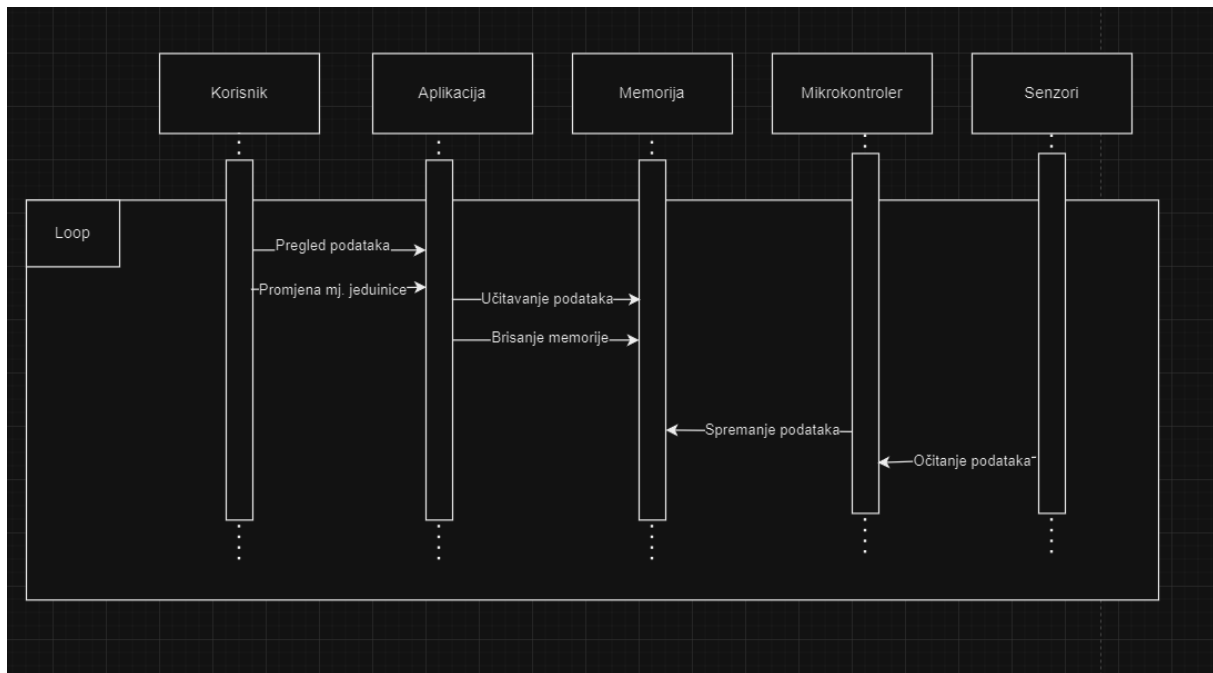
Slika 1. Dijagram slučajeva uporabe.

Slika 2. Prikazuje dijagram aktivnosti. Korisnik pri ulasku u web aplikaciju može pregledati vrijednosti senzora, ukoliko želi može promijeniti mjernu jedinicu i može obrisati povijest očitavanja vrijednosti.



Slika 2. Dijagram aktivnosti.

Slika 3. Prikazuje dijagram slijeda. Korisnik pregledava podatke iz web aplikacije i može promijeniti mjernu jedinicu. Aplikacija učitava podatke iz memorije i ima funkciju brisanja memorije. Mikrokontroler sprema podatke u memoriju, koji su očitani od senzora.



Slika 3. Dijagram slijeda.

### 3.1 Pretpostavke

Pretpostavljamo da će se projekt izvoditi u unutarnjem okruženju, te da će mikrokontroler imati pristup Wi-Fi mreži za komunikaciju s web aplikacijom.

### 3.2 Funkcijski zahtjevi

#### 3.2.1 Opća funkcionalnost

Zahtjev	Opis
F-1.1	Uređaj treba prikupljati podatke o temperaturi, vlažnosti zraka i svjetlosnoj jakosti u redovitim intervalima.
F-1.2	Web aplikacija mora imati gumb za brisanje povijesti.
F-1.2.1	Web aplikacija mora imati gumb za mijenjanje mjerne jedinice.

Tablica 2. Opća funkcionalnost.

#### 3.2.2 Funkcionalnosti načina rada

F-2.1	Uređaj će podržavati automatski način rada gdje će podaci biti prikupljeni automatski.
F-2	Uređaj će podržavati ručni način rada gdje će podaci biti prikupljeni na zahtjev korisnika.

---

*Tablica 3. Funkcionalnosti načina rada.*

### 3.2.3 Funkcionalnosti nadzornog načina rada

F-3.1	Kada je poslana naredba "TEMP" uređaj šalje podatke o trenutnoj temperaturi
F-3.2	Kada je poslana naredba "HUM" uređaj šalje podatke o trenutnoj vlažnosti zraka.
F-3.3	Kada je poslana naredba "LIGHT" uređaj šalje podatke o trenutnoj jačini svijetla.

---

*Tablica 4. Funkcionalnosti nadzornog načina rada.*

### 3.2.4 Zahtjevi napajanja

Zahtjev	Opis
F-4.1	Uređaj će se napajati preko USB priključka
F-4.2	Potrebno je koristiti energetske učinkovite komponente kako bi se smanjila potrošnja energije.
F-4.3	Mogućnost rada sustava na prijenosnoj bateriji srednje kvalitete.

*Tablica 5. Zahtjevi napajanja.*

## 3.3 Mehanički zahtjevi

### 3.3.1 Dimenzije i kućište

Zahtjev	Opis
F-4.1	Uređaj će biti kompaktnih dimenzija, prilagođenih smještaju senzora i mikrokontrolera.

## 3.4 Ostali zahtjevi

## 3.5 Cijena

Zahtjev	Opis
F-4.1	Cijena sustava ne smije biti veća od 400 kn

*Tablica 6. Cijena.*

### 3.6 Buduće značajke

Ove dodatne značajke se ne planiraju implementirati u prototipu. To su ideje za budući razvoj uređaja.

Zahtjev	Opis
O-8.1	Uređajem se može upravljati Bluetooth
O-8.2	Web aplikacija je dostupna izvan lokalne mreže. Pristup s veće udaljenosti.

*Tablica 7. Buduće značajke.*

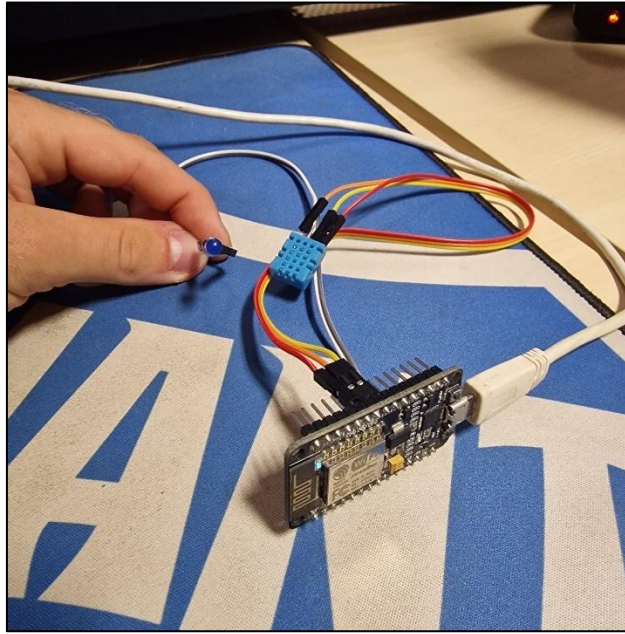
## 4 Detaljnije specifikacije funkcije

### 4.1 Implementacija i korisničko sučelje

#### 4.1.1 Oblikovanje sklopovlja

Arduino sustav spojimo na napajanje preko USB-a. Svi senzori se napajaju putem Arduina.

Slika 4. prikazuje razvojnu pločicu sa sensorima.



*Slika 4. Oblikovanje sklopa.*

#### 4.1.2 Sučelje sklopovlja

ESP8266 mikrokontroler ima ugrađeno USB sučelje koje omogućava komunikaciju s računalom. To USB sučelje naziva se "UART to USB Bridge", a često se koristi za programiranje mikrokontrolera, prijenos podataka i serijsku komunikaciju.

Kada spojite ESP8266 na računalo putem mikro USB kabela, ESP8266 se može ponašati kao virtualni serijski uređaj. To znači da se podaci mogu serijski prenositi između ESP8266 i računala koristeći standardne UART naredbe.

#### 4.1.3 Oblikovanje programske potpore

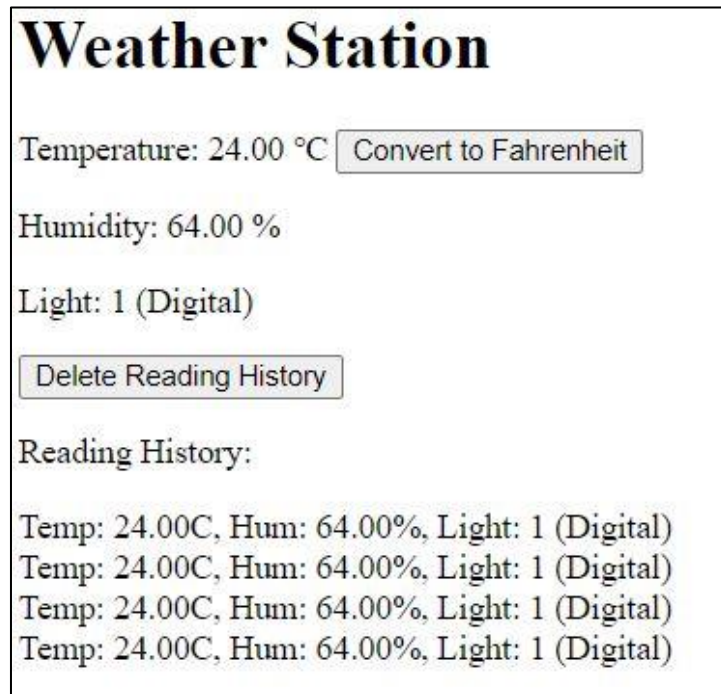
Prvi dio programske potpore je web aplikacija, na kojoj se korisniku ispisuju trenutna očitavanja temperature, vlage u zraku i svjetlosti. Također korisnik je u mogućnosti vidjeti i povijest očitavanja svih senzora u obliku tablice. Osim HTML-a, koristimo i malo JavaScripta koja omogućuje pretvorbu mjerne jedinice iz Celzijevih stupnjeva u Farenhajte i obrnuto.

Drugi dio programske potpore je C++ programski kod koji upravlja mikrokontrolerom i očitava podatke sa senzora, te ih prikazuje pomoću web aplikacije.



#### 4.1.4 Implementacija

Slika 5. prikazuje korisničko sučelje u obliku web aplikacije. Prikazuje se trenutno očitavanje svih senzora. Klikom na gumb „*Conver to Fahrenheit*“ prebacuje se prikaz mjernih jedinica. Također dostupna je i povijest svakog očitavanja i pritiskom na gumb „*Delete Reading History*“ povijest se briše.



Slika 5. Web aplikacija.

Na Slici 6. i 7. prikazan je „C++“ kod koji upravlja mikrokontrolerom.

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <DHT.h>

const char* ssid = "BowserFlash13";
const char* password = "aaa";

const char* PARAM_TEMP = "temp";
const char* PARAM_HUM = "hum";
const char* PARAM_LIGHT = "light";

const int DHTPin = D5;
const int LightPin = D0;

DHT dht(DHTPin, DHT22);

ESP8266WebServer server(80);

float temperature = 0;
float humidity = 0;
int lightValue = 0;
String dataLog = "";

void setup() {
  Serial.begin(115200);
  dht.begin();

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  pinMode(LightPin, INPUT); // Set the photoresistor pin as input

  server.on("/", HTTP_GET, [](void){
    String tempStr = String(temperature);
    String humStr = String(humidity);
    String lightStr = String(lightValue);
    String html = "<html><body>";
    html += "<h1>Weather Station</h1>";
    html += "<p>Temperature: " + tempStr + " °C</p>";
    html += "<p>Humidity: " + humStr + " %</p>";
    html += "<p>Light: " + lightStr + " (Digital)</p>";
    html += "<form action='/delete' method='post'><input type='submit' value='Delete Reading History'></form>";
    html += "<p>Reading History:</p>" + dataLog;
    html += "</body></html>";
    server.send(200, "text/html", html);
  });
};
```

Slika 6. Prvi dio programskog koda.

```

server.on("/delete", HTTP_POST, [](){
  dataLog = "";
  server.send(200, "text/plain", "Reading history deleted.");
});

server.begin();

Serial.println("Server started");
}

void loop() {
  server.handleClient();

  delay(10000);
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  lightValue = digitalRead(LightPin); // Read digital value from photoresistor

  String logEntry = "Temp: " + String(temperature) + "C, Hum: " + String(humidity) + "%, Light: " + String(lightValue) + " (Digital)<br>";
  dataLog += logEntry;
  Serial.println(logEntry);

  // Konzola
  if (Serial.available() > 0) {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command == "TEMP") {
      Serial.print("Current Temperature: ");
      Serial.print(temperature);
      Serial.println("C");
    } else if (command == "HUM") {
      Serial.print("Current Humidity: ");
      Serial.print(humidity);
      Serial.println("%");
    } else if (command == "LIGHT") {
      Serial.print("Current Light: ");
      Serial.println(lightValue);
    } else {
      Serial.println("Error");
    }
  }
}

```

Slika 7. Drugi dio programskog koda.

Na početku programskog koda se nalaze potrebne biblioteke. Učitavamo biblioteke Bežične mreže i servera, te učitamo biblioteku senzora.

Definiramo konstante kao ime mreže i zaporku i ostale varijable koje će nam kasnije biti potrebne. Zatim postavljamo sve varijable na početnu vrijednost. Uspostavljamo bežičnu vezu.

Oblikujemo jednostavnu web aplikaciju pomoću HTML-a, CSS-a i JavaScripta. Definiramo funkciju za brisanje povijesti očitavanja, koja su pohranjena u varijablu „dataLog“. Zatim u ponavljajućoj petlji očitamo vrijednosti od senzora, spremamo ih u zasebne varijable i u povijest očitavanja. Zadnji dio programskog koda je zaslužan za ispis vrijednosti senzora na zahtjev korisnika u konzoli. Ta petlja se ponavlja svakih 10 sekundi.

Na Slici 8. prikazana je konzola sustava. Periodično šalje očitavanja. Upisivanjem „TEMP“, sustav nam vraća trenutno očitavanje temperature, Sukladno i s naredbama „HUM“ i „LIGHT“

```
Connecting to WiFi...
Server started
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Current Temperature: 24.00C
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Current Humidity: 64.00%
Temp: 24.00C, Hum: 64.00%, Light: 1 (Digital)<br>
Current Light): 1
```

Slika 8. Ispis konzole.

## 5 Korištene tehnologije i alati

Za pisanje programsko koda je korišteno razvojno okruženje „Arduino IDE 1.8.13“, uz dodatno instalirane biblioteke za razvojnu pločicu ESP 8266 i ostale senzore.

Za pristup web aplikaciji potreban je web preglednik, u ovom slučaju „Google Chrome“ aplikacija je bila dostupna u lokalnom mreži na adresi „<http://192.168.1.5/>“.

Sama web aplikacija je pisana u HTML-u, uz malo CSS-a i JavaScripta koja se nalazi u programskom kodu.

Za izradu dijagrama je korištena web alat „draw.io“.

## 6 Zaključak i budući rad

Arduino Meteo Stanica predstavlja uspješnu realizaciju ciljeva postavljenih u projektnom zadatku. Projekt je temeljen na razvoju malene meteorološke stanice koja prikuplja podatke o temperaturi, vlažnosti zraka i svjetlosnoj jakosti te ih prikazuje putem web aplikacije. Kroz provedene faze projektiranja, implementacije te testiranja, ostvarena je funkcionalnost sustava koja omogućava korisnicima da prate trenutne vremenske uvjete na odabranoj lokaciji.

U sklopu projekta korištene su različite tehnologije i alati poput ESP8266 mikrokontrolera, DHT11 senzora za mjerenje temperature i vlažnosti zraka te fotoresistora za mjerenje svjetla. Web aplikacija je implementirana pomoću HTML-a, CSS-a i JavaScripta te omogućava korisnicima da vizualiziraju prikupljene podatke i prate njihovu povijest. Osim toga, implementirana je i mogućnost interakcije s uređajem putem Arduino IDE konzole.

Projekt je ostvario postavljene funkcionalne zahtjeve, kao što su automatski i ručni način rada, praćenje svjetlosne jakosti te komunikacija putem konzole. Minimalni zahtjevi u smislu točnosti podataka, stabilnosti rada te korisničkog sučelja također su zadovoljeni.

Projekt je omogućio stjecanje znanja o razvoju ugradbenih sustava, programiranju mikrokontrolera, senzora te implementaciji web aplikacije. Također, kroz projekt su se razvijale vještine rada s različitim komponentama, rješavanje tehničkih izazova te razumijevanje interakcije između hardverskog i softverskog dijela.

U budućem razvoju projekta moguće je razmatrati dodatne značajke kao što su upravljanje uređajem putem Bluetootha, širenje dostupnosti web aplikacije izvan lokalne mreže te optimizacija energetske potrošnje. Ovaj projekt otvara vrata daljnjem istraživanju i primjeni u različitim područjima kao što su poljoprivreda, kućna automatizacija te slično.

## 7 Reference

Internet:

- [1] DTH 11 Senzor - <https://components101.com/sensors/dht11-temperature-sensor>
- [2] ESP 8266 - <https://randomnerdtutorials.com/esp8266-web-server/>
- [3] Fotorezistor - <https://www.instructables.com/Light-Sensor-Photoresistor-Arduino-Tinkercad/>