

# NRF24 modul za komunikaciju između mikroupravljača

---

Tandara, Marko

Undergraduate thesis / Završni rad

2020

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:477033>

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

*Download date / Datum preuzimanja:* **2024-11-23**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**NRF24 MODUL ZA KOMUNIKACIJU IZMEĐU  
MIKROUPRAVLJAČA**

**Završni rad**

**Marko Tandara**

**Osijek, 2020.**

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
<b>2. BEŽIČNA KOMUNIKACIJA S NRF24 TEHNOLOGIJOM</b> .....	<b>2</b>
<b>2.1. Radio komunikacija</b> .....	<b>2</b>
<b>2.2. NRF24</b> .....	<b>2</b>
<b>2.3. Topologija stabla NRF24 modula</b> .....	<b>3</b>
<b>3. SKLOPOVLJE I PROGRAMSKA PODRŠKA</b> .....	<b>5</b>
<b>3.1. Sklopovlje</b> .....	<b>5</b>
3.1.1. Arduino .....	5
3.1.2. Arduino Nano .....	6
3.1.3. Wemos D1 mini .....	7
3.1.4. OLED zaslon .....	8
3.1.5. Vrste komunikacijskih protokola .....	9
3.1.6. SPI .....	10
3.1.7. I2C .....	11
<b>3.2. Programska podrška</b> .....	<b>12</b>
3.2.1. Kreiranje objekata i adresa .....	13
3.2.2. Povezivanje na Wifi mrežu .....	13
3.2.3. Dohvaćanje podataka putem API poziva .....	13
3.2.4. Slanje podataka putem NRF24 modula .....	14
3.2.5. Ispis podataka na OLED zaslon .....	14
3.2.6. Primanje podataka putem NRF24 modula .....	15
<b>3.3. Razvijeni uređaji</b> .....	<b>16</b>
<b>3.4. Testiranje</b> .....	<b>17</b>
<b>4. ZAKLJUČAK</b> .....	<b>20</b>
<b>LITERATURA</b> .....	<b>21</b>
<b>SAŽETAK</b> .....	<b>22</b>
<b>ABSTRACT</b> .....	<b>23</b>
<b>ŽIVOTOPIS</b> .....	<b>24</b>
<b>PRILOZI</b> .....	<b>25</b>

# 1. UVOD

Zadatak ovog rada je napraviti mrežu između najmanje tri NRF24 modula, gdje će jedan mikroupravljač predstavljati glavnu jedinicu. On će biti spojen na Wemos D1 mini mikroupravljač koji će se spojiti na zadanu internetsku mrežu i prikupljati podatke o trenutnoj temperaturi i vlazi za grad Osijek putem API (*engl. Application Programming Interface*) poziva s web stranice, te će parsirane podatke iz JSON (*engl. JavaScript Object Notation*) objekta ispisivati na OLED (*engl. organic light emitting diode*) zaslon. Te podatke će putem NRF24L01+ modula slati na druga dva sklopovlja bazirana na Arduino Nano razvojnoj pločici, koji će isto putem NRF24L01+ modula primiti te podatke i ispisivati ih na OLED zaslonu. Prijenos podataka između dva NRF24L01+ modula se odvija preko radio valova. Prijenos podataka je podešen na 2 Mbs. Komunikacija između NRF24L01+ modula i nekog mikroupravljača odvija se putem SPI (*engl. Serial Peripheral Interface*) komunikacijskog protokola, dok se komunikacija između Arduino Nano-a ili Wemos D1 mini-a sa OLED zaslonom ostvaruje putem I2C (*engl. Inter-Integrated Circuit*) komunikacijskog protokola. Glavna jedinica će svakih 10 sekundi slati podatke prijemnim jedinicama, a prijemne jedinice će slati potvrdu o uspješnoj transmisiji podataka, što će glavna jedinica koristiti kao uvjet za ispis adrese prijemnika na OLED zaslonu pokraj podatka kojeg je poslala. Ako prijemna jedinica ne primi nikakve podatke unutar 30 sekundi od uključivanja ili zadnje komunikacije, na OLED zaslon će ispisati upozorenje, da komunikacija s glavnom jedinicom nije ostvarena. Ostatak rada je organiziran na sljedeći način. U poglavlju 2 opisan je NRF24 modul i njegov način komuniciranja. U poglavlju 3 opisana je izrada sklopovlja (obilježja korištenih komponenti) i programske podrške te njihovo testiranje.

## 1.1. Zadatak završnog rada

U radu je potrebno opisati načine komunikacije i povezivanja mikroupravljača s modulom NRF24. Potrebno je provesti testiranje komunikacije s najmanje tri mikroupravljačka sustava od kojih je jedan glavni.

## 2. BEŽIČNA KOMUNIKACIJA S NRF24 TEHNOLOGIJOM

### 2.1. Radio komunikacija

Za ostvarivanje radio prijenosa potrebni su odašiljač i prijemnik. Odašiljač protjecanjem izmjenične struje u anteni stvara signal s informacijom koji šalje kroz prostor do antene prijemnika, koji prima signal te dohvaća informacije iz tog signala.

Sam proces stvaranja i slanja signala s informacijom se odvija u nekoliko koraka. Odašiljač sadrži oscilator koji stvara signale jednake i vrlo visoke stabilne frekvencije s konstantnom amplitudom. Takav signal je nosioc, što znači da ne sadrži nikakvu informaciju pa je potrebno modulatorom utisnuti informaciju u takav signal postupkom zvanim modulacija. Modulacijom se sinusnom signalu nosiocu (koji je na višoj frekvenciji nego signal ili informacija koju prenosimo) mijenjaju neki parametri (frekvencija, faza ili amplituda), kako bi se stvorio novi signal koji sadrži informaciju.

Primjeri modulacija:

- FSK (*engl. Frequency Shift Keying*) modulacija – mijenja se signalova frekvencija
- ASK (*engl. Amplitude Shift Keying*) modulacija – mijenja se signalova amplituda
- PSK (*engl. Phase Shift Keying*) modulacija – mijenja se signalova faza
- QAM (*engl. Quadrature Amplitude Modulation*) – mijenja se signalova faza i amplituda.

Signal dobiven modulacijom je nepogodan za odašiljanje, jer je vrlo male snage, te ga je potrebno dovesti na pojačalo snage. Cijeli ovaj postupak se obavlja kako bi dobili učinkovit signal, odnosno da je spektralna učinkovitost vrlo velika, što znači da signal neće zauzimati velik dio spektra. Signal spreman za slanje šalje se prijemniku putem antene, te se na prijemniku odvija obrnut proces, odnosno proces demodulacije, gdje se informacija izdvaja iz poslanog signala [1].

### 2.2. NRF24

U ovom radu će se koristiti NRF24L01 modul namijenjen za bežičnu komunikaciju prikazan na slici 2.1. Omogućava dvosmjernu komunikaciju, što znači da može slati i primiti podatke u isto vrijeme. Ovaj modul je jako pogodan za izradu sklopovlja baziranih na bežičnoj komunikaciji zbog svoje jednostavnosti, niske potrošnje, niske cijene te velike učinkovitosti.

NRF24L01 primopredajnik je dizajniran da radi na širokopojasnom ISM (*engl. Industrial, Scientific, Medical networks*) frekvencijskom rasponu i koristi diskretnu GFSK modulaciju za

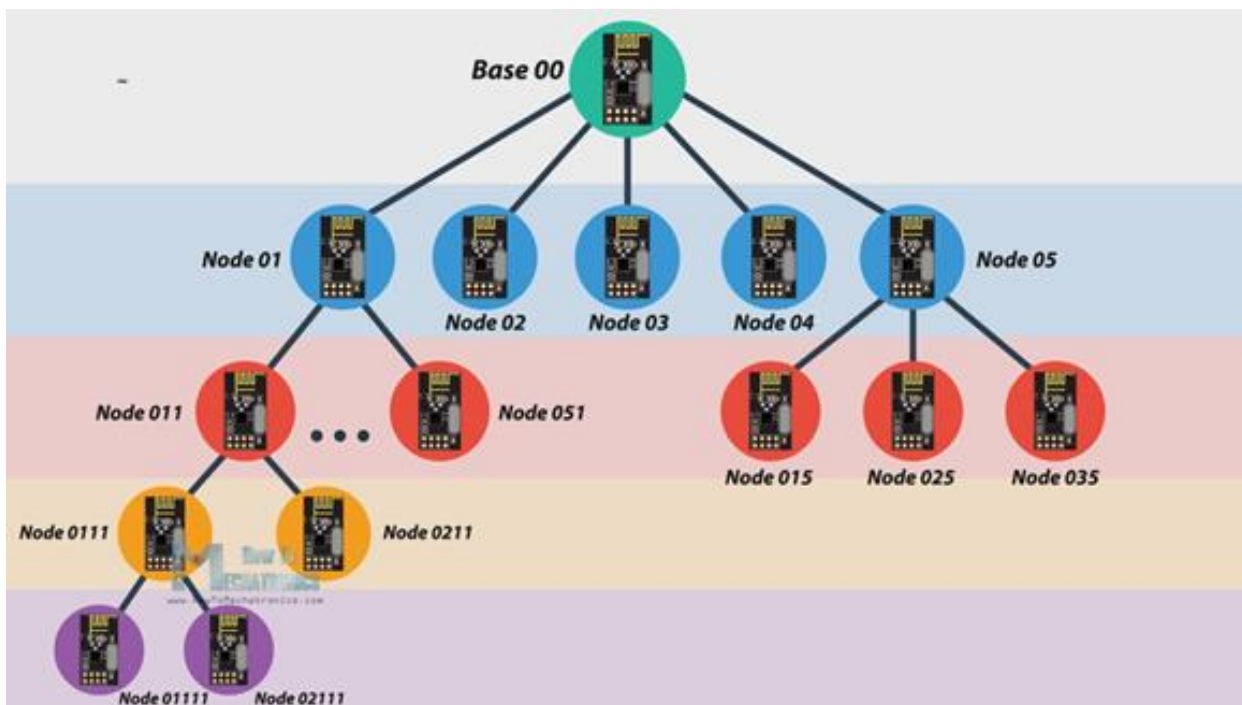
odašiljanje podataka. Frekvencijski pojas se kreće od 2.400 do 2.525 GHz, što omogućava rad na 125 kanala. Kako bi dva ili više modula komunicirala jedan s drugim moraju se nalaziti na istom kanalu. Kod GFSK modulacije se kao parametar signala nosioca mijenja frekvencija, kao kod FSK modulacije, samo što je kod GFSK modulacije signal koji sadrži informaciju proveden kroz Gaussov filter, te se tom signalu smanjuju nagibi promjene stanja iz logičke 0 u logičku 1, a tako i spektar. Tim postupkom, promjena frekvencije se ne događa odmah, nego postepeno, što omogućava zauzimanje manjeg dijela spektra [2]. Brzina prijenosa podataka se kreće od 250 Kbps do 2 Mbps. Ako se koriste manje brzine modul može komunicirati na udaljenostima do 100 m s ugrađenom antenom. Postoji verzija s posebnim pojačalom i antenom koja može komunicirati na udaljenostima do 1 km. Primopredajnik radi na naponima od 1.9 do 3.6 V, s potrošnjom struje od 12 mA u stanju mirovanja. Kada odašilje podatke potrošnja doseže 115 mA, a kada prima do 45 mA. Za konfiguraciju i upravljanje ovog modula koristi se SPI komunikacija [3].



Sl. 2.1. NRF24L01 modul sa konektorima [3].

### 2.3. Topologija stabla NRF24 modula

U mreži NRF24L01 primopredajnici su povezani adresama dodijeljenim kanalima (*engl. pipes*). Svaki modul može slušati do 6 drugih modula putem 6 kanala. Ova ograničenost se može zaobići korištenjem „NRF24Network“ biblioteke, tako što svaki modul ima jednog roditelja i 5 djece, stvarajući mrežnu topologiju stabla. Ovo stablo je ograničeno dubinom od 5 roditelja. Što ograničava broj mogućih modula u mreži na 3125 ( $5^5$ ). Moduli u ovoj topologiji komuniciraju direktno s roditeljem i djecom. Omogućeno je komuniciranje svih modula u mreži. Svaki modul u ovoj topologiji mora biti adresiran 15 bitnom adresom, koju je najbolje zapisivati u oktalnom zapisu. Na slici 2.2 može se primijetiti kada modul 011 želi komunicirati s modulom 02, komunikacija se odvija kroz module 01 i 00, što znači da ovi moduli moraju biti aktivni cijelo vrijeme dok komunikacija ne završi [4].



Sl. 2.2. Topologija stabla [5].

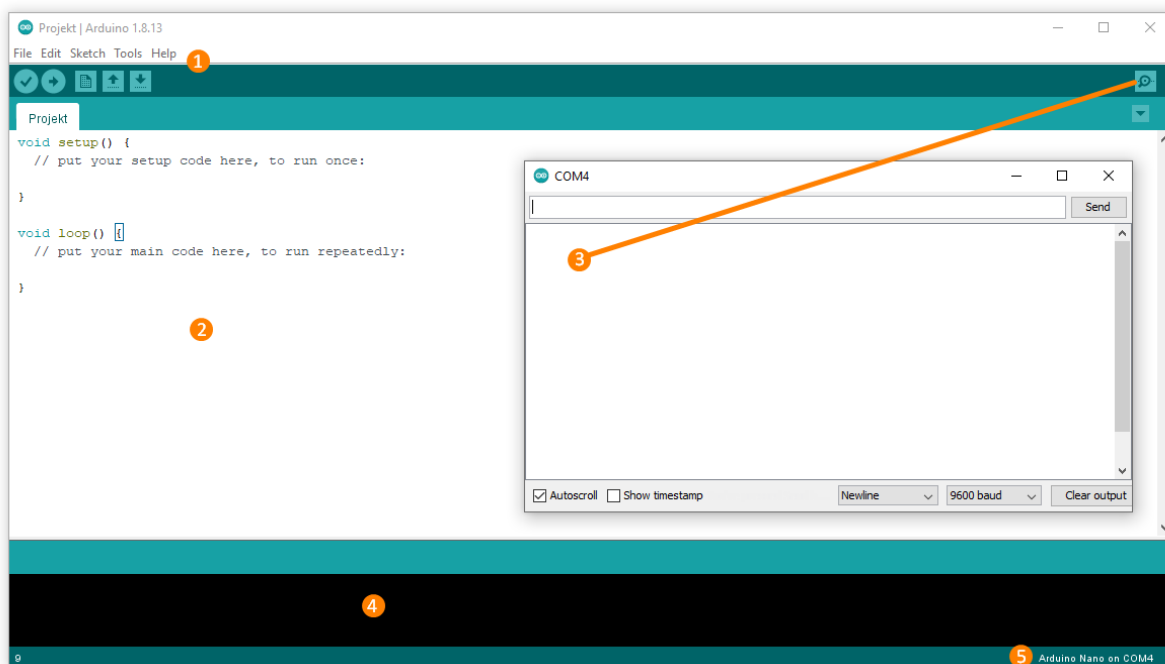
## 3. SKLOPOVLJE I PROGRAMSKA PODRŠKA

### 3.1. Sklopovlje

#### 3.1.1. Arduino

Arduino platforma se koristi za izgradnju elektroničkih projekata koji međusobno komuniciraju. Otvorenog je tipa (*engl. open-source*), što znači da je dozvoljeno dijeljenje podataka o programskoj podršci i sklopovljima koje je dopušteno izmjenjivati i nadograđivati. Isto tako dozvoljeno je stvaranje novih razvojnih pločica i razvojnih okruženja baziranih na Arduino platformi.

Za pisanje, kompajliranje i učitavanje programskog kôda na Arduino mikroupravljače koristi se Arduino IDE (*engl. Integrated Development Environment*) razvojno okruženje (Slika 3.1). Programski kôd u Arduino IDE piše se programskim jezikom deriviranim iz C/C++ programskog jezika. Mikroupravljač za kojeg se piše programski kôd mora na sebi imati već prethodno isprogramiran *Bootloader*, odnosno dio kôda koji će primati podatke putem UART (*engl. Universal Asynchronous Receiver-Transmitter*) komunikacije i tako programirati samog sebe. On se isto prvi pokreće prilikom uključivanja razvojne pločice, te dohvaća kôd iz memorije i pokreće njegovo izvršavanje [6].



Sl. 3.1. Izgled Arduino IDE sučelja.



Neki od glavnih dijelova Arduino IDE su:

1. Izbornička traka – na njoj se mogu otvarati pojedini izbornici u kojima se nalaze brojne funkcionalnosti kao što su primjeri kôdova, upravljanje editorom, upravljanje bibliotekama, odabiranje porta, itd.
2. *Editor* – Prostor namijenjen za pisanje programskog kôda
3. Serijski monitor – Prozor koji se otvara kada se pritisne gore desno označeni gumb. Služi za ispis podataka koje mikroupravljač šalje putem UART komunikacije. Kako bi komunikacija ispravno radila potrebno je imati istu brzinu prijenosa, odnosno *baud rate*
4. Informacijska traka – Na njoj se očitavaju trenutna stanja i greške prilikom kompajliranja ili učitavanja kôda na mikroupravljač. Veoma korisno kod lociranja grešaka u napisanom kôdu
5. Traka za prikazivanje trenutno odabranih razvojnih pločica i portova za učitavanje kôda.

Na slici 3.1 vidljivo je da kada se pokreće projekt u *Editoru* se pojavljuju 2 funkcijska bloka (*void setup* i *void loop*). Kod pokretanja mikroupravljača prvo se izvodi funkcija *void setup* i ona se izvršava samo jedanput, zatim se pokreće funkcija *void loop*, koja sadrži glavni dio programa koji obavlja neku radnju. *Void loop* funkcija se neprestano ponavlja, što znači ako je izvršen sav program u njoj funkcija će se ponovno pokrenuti i tako u nedogled. Svaki se program mora sastojati od te dvije funkcije.

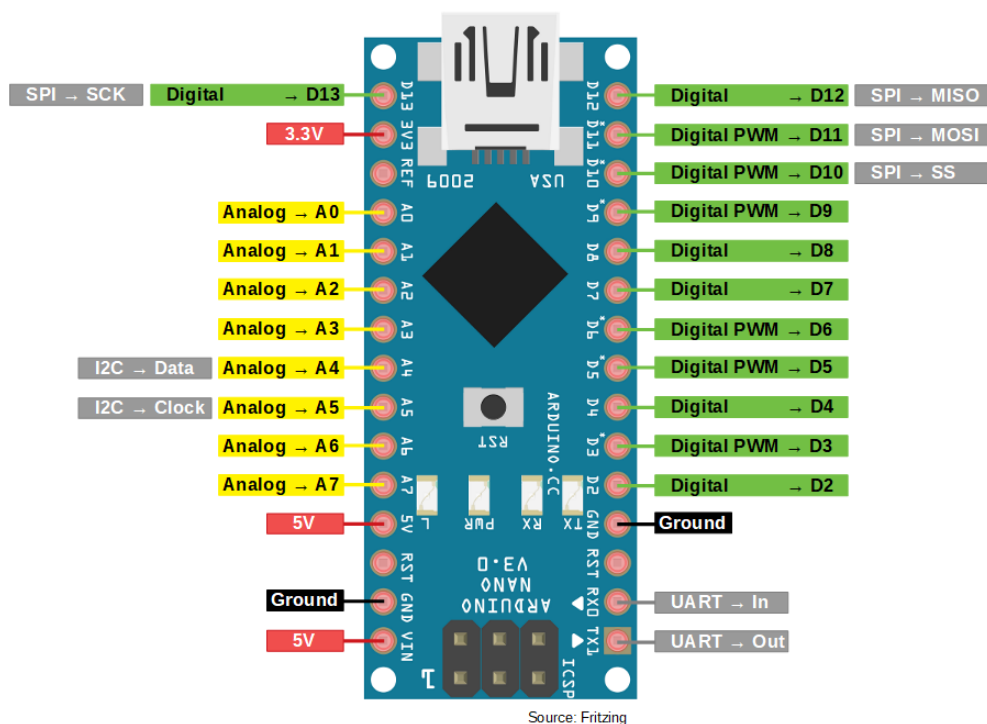
### 3.1.2. Arduino Nano

U ovom radu korištene su Arduino Nano v3.0 razvojne pločice (Slika 3.2). One su bazirane na Atmega328P mikroupravljaču od Atmela. Arduino Nano jedna je od najučinkovitiji i najjeftinijih razvojnih pločica Arduino okruženja. Za napajanje i učitavanje kôda na Atmega328P se koristi Mini-B USB priključak. Pločica sadrži 14 digitalnih ulazno/izlaznih nožica. Vrijednosti koje se šalju putem tih nožica mogu biti *HIGH* ili *LOW*, ovisno šalje li se na nožicu 1 ili 0. Struja za ulazno/izlazne nožice iznosi 40 mA. U tablici 3.1 su prikazana neka obilježja Atmega328P mikroupravljača.

Tab. 3.1. Obilježja Atmega328P.

<b>Mikroupravljač</b>	Atmega328P
<b>Flash Memorija</b>	32 KB (2 KB za <i>Bootloader</i> )
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1024 B
<b>Brzina sata</b>	16 MHz

Od 14 digitalnih nožica 6 se može koristiti kao PWM (*engl. Pulse Width Modulation*) izlaz. PWM ili pulsno-širinska modulacija je metoda dobivanja analognih vrijednosti pomoću digitalnih impulsa konstantne amplitude. Arduino Nano također ima 8 analognih nožica gdje se koristi 10 bitni ADC (*engl. Analog to Digital Converter*). AD pretvornik pretvara ulazni napon od 0 do 5 V u vrijednosti između 0 i 1023. Arduino Nano ima jednu od 3.3 V i dvije od 5 V naponske nožice od kojih je jedna VIN nožica. S VIN nožicom ulazni napon može biti od 7 do 12 V koja se koristi za vanjski izvor napajanja. 5V nožica pruža regulirano napajanje mikroupravljača i drugih komponenti. 3.3 volta na nožici je generirano od ugrađenog naponskog regulatora na pločici. Sve tri nožice imaju protok struje od 50 mA. Također ima GND nožicu koja služi za uzemljenje [7].



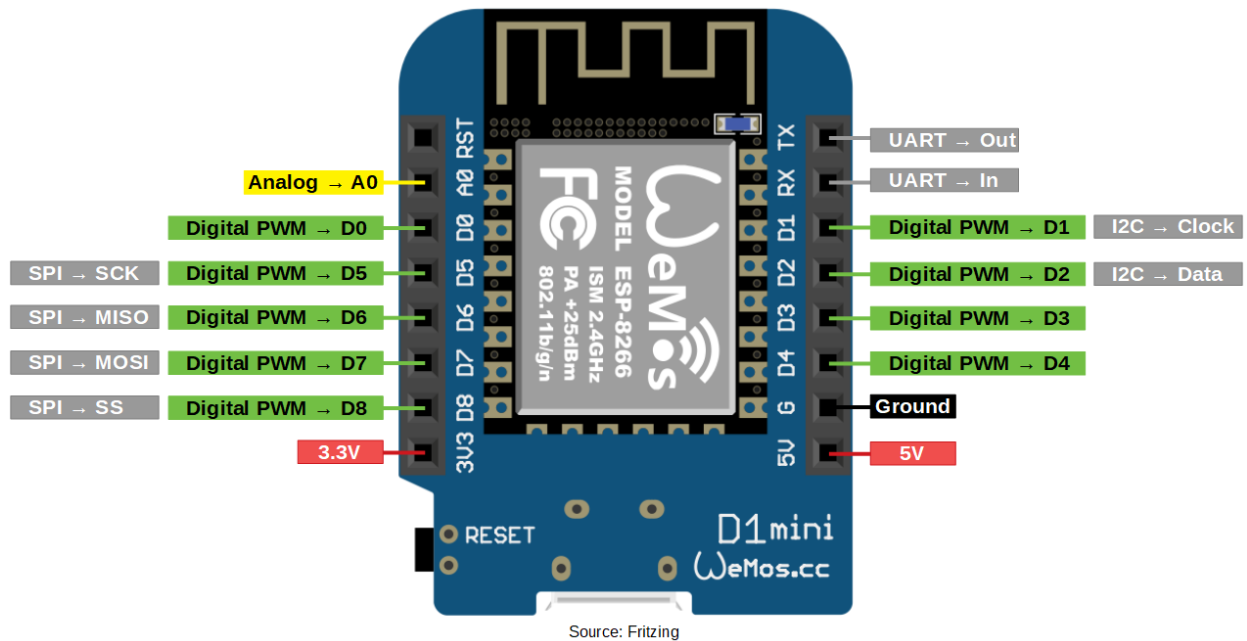
Sl. 3.2. Arduino Nano v3.0 [7]

Sadrži tipku reset za ponovno pokretanje mikroupravljača. Rx i Tx nožice služe za slanje i primanje TTL (*engl. Transistor-Transistor Logic*) serijskih podataka. Bitne karakteristike koje ima a bitne su za ovaj rad su SPI i I2C komunikacijski protokoli koji će se koristiti za ispis podataka na OLED zaslon i komunikaciju putem NRF24 modula.

### 3.1.3. Wemos D1 mini

Wemos D1 mini je razvojna pločica bazirana na ESP8266 mikroupravljaču (Slika 3.3). Bitna karakteristika koja ju razlikuje od ostalih razvojni pločica kao što su Arduino Nano je ta da na

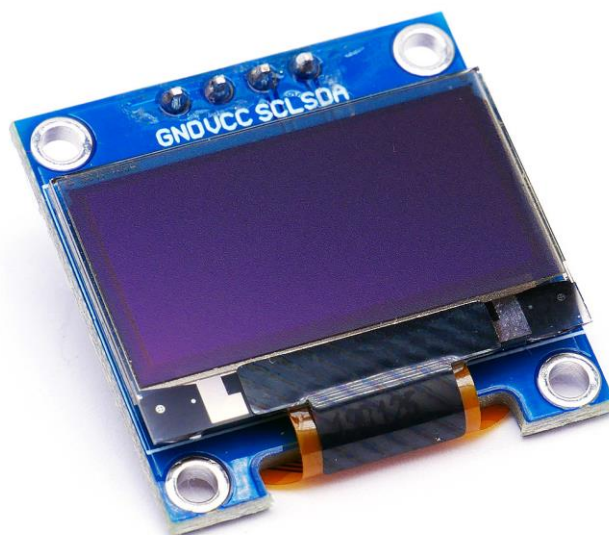
mikroupravljaču sadrži Wifi modul. Napaja se pomoću 5 V micro USB priključka. *Flash* memorija ESP8266 iznosi 4 MB. Wemos D1 Mini ima 1 analognu nožicu i 11 digitalnih. Sve nožice rade na naponu od 3.3 V. Također ima nožice od 3.3 i 5 V za napajanje drugih komponenti, a isto se mogu koristiti za vanjsko napajanje. Obje nožice pružaju 500 mA. Wemos D1 mini podržava protokole za serijsku komunikaciju kao što su SPI i I2C [8].



Sl. 3.3. Wemos D1 mini [8].

### 3.1.4. OLED zaslon

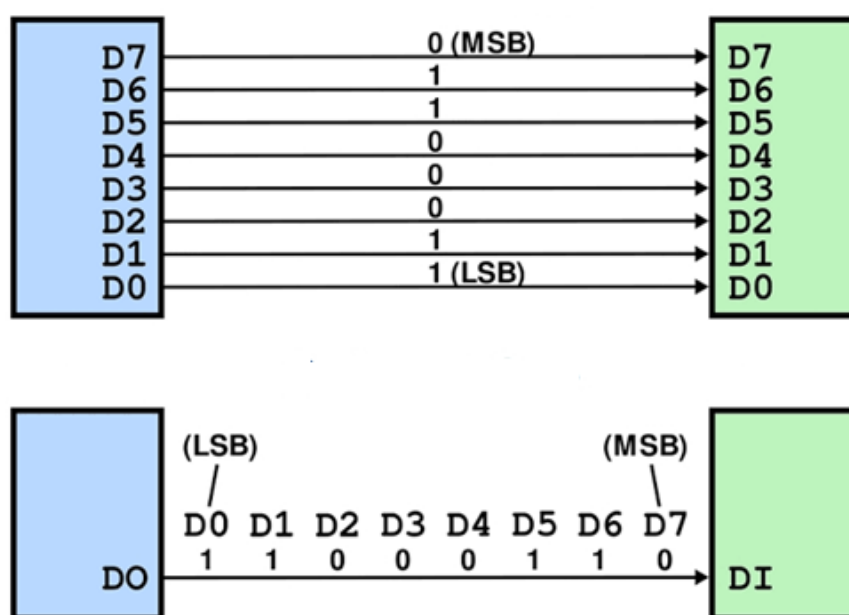
Za prikazivanje obrađenih podataka koristi se jednobojni OLED zaslon SSD1306, dijagonale 2.4384 cm i rezolucije 128x64 piksela, prikazan na slici 3.4. Zaslon se napaja naponom od 3.3 V ali zbog već ugrađenog regulatora napona, moguće ga je spojiti na napajanja do 5 V, što ga čini kompatibilnim s većinom arduino razvojnih pločica. Ovi zaslone koriste SPI ili I2C komunikacijske protokole, neki mogu koristiti čak oba. Zaslone korišteni u ovom radu koriste I2C serijsku komunikaciju. Modul sadrži 4 nožice: VCC, GND, SCL i SDA. VCC služi za napajanje zaslona. GND služi za uzemljenje zaslona. SCL i SDA služe za serijsku komunikaciju putem I2C protokola i spajaju se na SCL i SDA nožice neke druge razvojne pločice [9]. Optimalna potrošnja ovih zaslona je 20 mA, ovisi o tome koliko je piksela na zaslonu uključeno, što ga čini učinkovitijim od ostalih LED zaslona na tržištu.



Sl. 3.4. OLED zaslon [10].

### 3.1.5. Vrste komunikacijskih protokola

Razlikuju se dvije vrste komunikacija za prijenos podataka putem komunikacijskog kanala. To su serijska i paralelna komunikacija. Kada koristimo paralelnu komunikaciju, svaki bit podatka se šalje u isto vrijeme, što zahtjeva više komunikacijskih linija. Kod serijske komunikacije bitovi podatka se šalju jedan za drugim putem jedne komunikacijske linije. Način rada obje komunikacije se može vidjeti na slici 3.5. Serijska komunikacija je znatno više zastupljenija u današnje vrijeme zbog svoje učinkovitosti. U nastavku će biti objašnjeno dva serijska komunikacijska protokola bitna za ovaj rad.



Sl. 3.5. Primjer paralelne i serijske komunikacije [11].

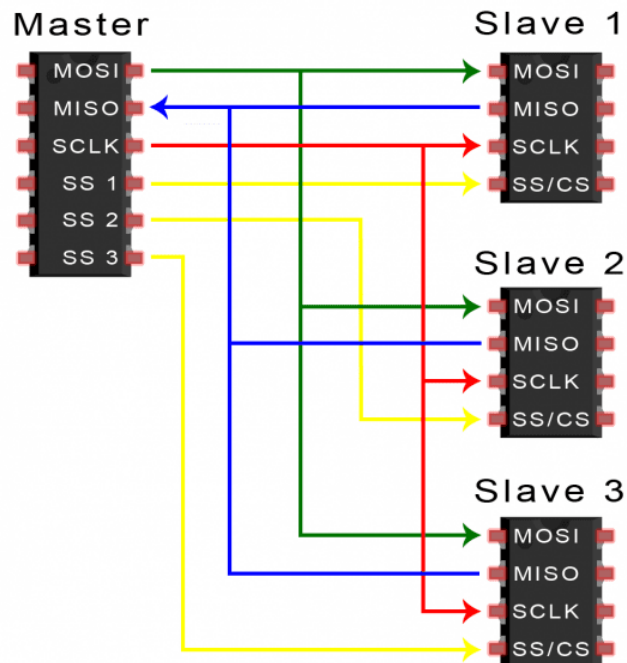
### 3.1.6. SPI

SPI je sinkroni serijski komunikacijski protokol korišten od strane mikroupravljača za brzu komunikaciju s jednim ili više perifernih uređaja. Kada se koristi SPI komunikacija uvijek postoji jedan uređaj koji je glavna jedinica koja upravlja drugim prijemnim odnosno perifernim uređajima što se može vidjeti na slici 3.6. Veza između glavne jedinice i perifernih uređaja je punog dupleksa, što znači da glavna jedinica može slati podatke na prijemne jedinice, a isto tako prijemne jedinice mogu slati podatke na glavnu jedinicu. Ranije je rečeno da je to sinkroni protokol, što znači da su podatci prilikom slanja sinkronizirani s linijom sata (engl. *Clock*). Putem *Clock* signala predajnik govori prijammiku kada treba uzrokovati bitove na podatkovnoj liniji. Zahvaljujući *Clock* signalu brzine prijenosa podataka na krajnjim uređajima komunikacijskog kanala ne moraju biti iste.

Kako bi ostvarili uspješnu komunikaciju između uređaja, potrebne su tri linije koje svi uređaji u mreži koriste zajedno i jedne linije koja je jedinstvena za svaki uređaj u mreži. Radi se o sljedećim linijama:

- MISO – (engl. *Master In Slave Out*) je ulazna nožica za glavnu jedinicu i izlazna za prijemnu jedinicu, odnosno prijenos podataka s prijemne jedinice prema glavnoj jedinici
- MOSI – (engl. *Master Out Slave In*) je ulazna nožica za prijemnu jedinicu a izlazna za glavnu, odnosno prijenos podataka s glavne jedinice prema prijemnoj jedinici
- SCK (engl. *Serial Clock*) – odgovorna za sinkroniziranje prijenosa podataka od strane pošiljatelja.

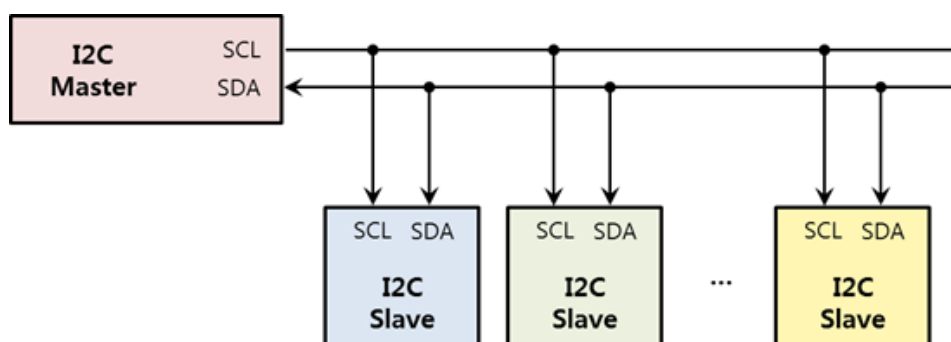
Postoji još jedna linija SS (engl. *Slave Select*), koju glavna jedinica koristi da informira prijemnu jedinicu hoće li joj slati podatke ili ne. Kada je ova nožica u *LOW* stanju, glavna jedinica obavještava prijemnika da šalje podatke, u suprotnom ako se ne šalju nikakvi podatci ova nožica bi trebala biti u *HIGH* stanju [12].



Sl. 3.6. Model nezavisne SPI komunikacije [13].

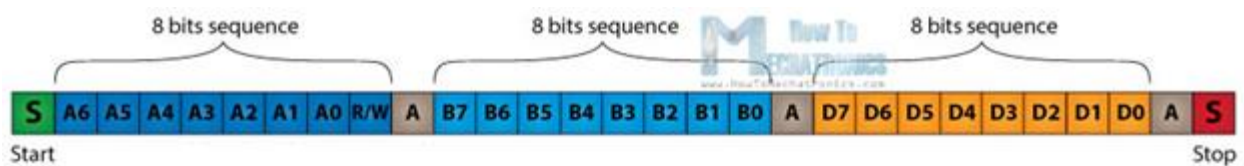
### 3.1.7. I2C

I2C je serijski komunikacijski protokol koji omogućava komunikaciju između jedne ili više glavnih jedinica s jednom ili više prijemnih jedinica. Za razliku od SPI protokola, za komunikaciju koristi samo dvije linije za slanje i primanje podataka vidljivo na slici 3.7. S pomoću te dvije linije moguće je ostvariti mrežu od 128 uređaja (kada koristimo 7 bitno adresiranje) ili 1024 (kada koristimo 10 bitno adresiranje). Jedna od tih linija je SCL (*engl. Serial Clock*) a druga SDA (*engl. Serial Data*). SCL se koristi za slanje *Clock* signala, a SDA za prijenos podataka.



Sl. 3.7. Model I2C komunikacije [15].

Svaki uređaj u I2C komunikaciji mora imati jedinstvenu 7-bitnu adresu, kako bi glavna jedinica znala s kojim uređajem treba komunicirati. Podatci putem SDA linije se šalju u sekvenci od 8 bitova. Promjena napona s 1 na 0 na SDA liniji, dok je SCL na 1 označava se *Start* bit odnosno početak prijenosa podataka od strane *mastera*. Zatim slijedi 8 bitni podatak s adresom uređaja s kojim *master* želi komunicirati. Nakon svakih 8 bitova slijedi ACK (*eng. Acknowledge*) bit. Nakon prvog ACK bita dolazi druga adresa koja označava broj registra na uređaju iz koje *master* želi čitati ili zapisivati podatke. Te nakon te adrese slijede 8 bitni podatkovni paketi. Nakon što su svi paketi poslani slijedi *Stop* bit, odnosno promjena napona s 0 na 1 na SDA liniji dok je SCL na 1, koja označava kraj prijenosa podataka. Cijeli taj proces je prikazan na slici 3.8 [14].



Sl. 3.8. Sekvenca bitova tijekom I2C komunikacije [14].

## 3.2. Programska podrška

Programski kôd za sva 3 mikroupravljača je pisan koristeći Arduino IDE razvojno okruženje. Kôd na Wemos D1 mini pločici se bitno razlikuje od kôda na prijemnim jedinicama, dok prijemne jedinice imaju gotovo identičan kôd, koji se samo razlikuje u adresama i podacima koje ispisuju na OLED zaslonu. Pri izradi programskog kôda korištene su razne biblioteke, koje su preuzete putem Arduino IDE. Radi se o sljedećim bibliotekama:

- <ESP8266WiFi.h> - Biblioteka koja omogućava rad s ESP8266 modulom
- <SPI.h> - Biblioteka za rad sa SPI komunikacijskim protokolom
- <RF24Network.h> <RF24.h> - Biblioteka za rad s NRF24L01+ modulom
- <ArduinoJson.h> - Biblioteka za rad s API pozivom
- <U8g2lib.h> - Biblioteka za rad s OLED zaslonom.

U narednom dijelu rada, bit će opisani važniji dijelovi programskog kôda glavne jedinice (Wemos D1 mini), a cijeli kôd se može pronaći u prilogu koji je na CD-u.

### 3.2.1. Kreiranje objekata i adresa

Pomoću uvedenih biblioteka mogu se stvoriti objekti na kojima se mogu pozivati funkcije tih biblioteka. Zatim se određuju adrese za pojedine jedinice u mreži (Slika 3.9). Adrese su zapisane u oktalnom formatu kako bi se znalo gdje im je pozicija u mreži baziranoj na topologiji stabla.

```
WiFiClient client;
RF24 radio(D3, D4);
RF24Network network(radio);
const uint16_t wemos = 00;
const uint16_t nano01 = 01;
const uint16_t nano02 = 02;
```

Sl. 3.9. Kreiranje objekata i zapis adresa.

### 3.2.2. Povezivanje na Wifi mrežu

Glavna jedinica se prvo mora spojiti na internetsku mrežu kako bi mogla dohvatiti podatke s interneta (Slika 3.10). Iz kôda se vidi da se mikroupravljač pokušava spojiti na Wifi mrežu prema korisničkom imenu i lozinki iz *header* datoteke. Ukoliko prođe 30 sekundi da konekcija nije uspješna program se ponovno pokreće. Ako je konekcija uspješna na serijski monitor se ispisuju podatci Wifi mreže na koju je mikroupravljač spojen i program nastavlja s radom.

```
WiFi.begin(ssid, password);
int counter = 0;
while (WiFi.status() != WL_CONNECTED)
{
    delay(300);
    if (counter++ > 100)
        ESP.restart();
    Serial.print( "." );
}
Serial.println("\nSuccessfully connected to WiFi");
printWifiStatus();
```

Sl. 3.10. Povezivanje na Wifi mrežu.

### 3.2.3. Dohvaćanje podataka putem API poziva

Pravi se HTTP (*engl. HyperText Transfer Protocol*) zahtjev s API pozivom, koji je napravljen na web stranici „[www.openweathermap.org](http://www.openweathermap.org)“. Nakon što je dobiven odgovor od web stranice, iz njega se uzima samo JSON datoteka u kojoj se nalaze podatci koji su potrebni za ovaj rad. Prvo se alocira memorija potrebna za instanciranje JSON objekta. Kada je stvoren objekt sa svim JSON podacima, iz njega se uzimaju podatci koji su potrebni za sustav (temperatura i vlaga) i spremaju se u lokalne varijable (Slika 3.11).



```

const size_t capacity = JSON_ARRAY_SIZE(1) + JSON_OBJECT_SIZE(1) +
2*JSON_OBJECT_SIZE(2) + JSON_OBJECT_SIZE(4) + 2*JSON_OBJECT_SIZE(5) +
JSON_OBJECT_SIZE(13) + 270;
DynamicJsonDocument doc(capacity);

DynamicJsonDocument doc(capacity);
DeserializationError error = deserializeJson(doc, client);
if (error) {
  Serial.print(F("deserializeJson() failed: "));
  Serial.println(error.c_str());
  return;
}
int weatherTemperature = doc["main"]["temp"].as<float>();
int weatherHumidity = doc["main"]["humidity"].as<int>();

```

*Sl. 3.11. Alociranje memorije za JSON objekt i njegovo parsiranje.*

### 3.2.4. Slanje podataka putem NRF24 modula

Nakon što su vrijednosti spremljene u varijable, potrebno ih je poslati prijemnim jedinicama u mreži. Podatke o temperaturi šalju se na jednu adresu, a podatke o vlazi na drugu. Funkcija za slanje tih podataka se poziva svakih 10 sekundi. Ako su podatci uspješno primljeni od strane prijemnih jedinica, one šalju potvrdu koja se sprema u varijable na kojima je poziv napravljen. Te varijable služe kao potvrde o uspješnosti komunikacije koje se koriste u funkciji za ispisivanje adresa prijemnih jedinica na zaslonu glavne jedinice. Primjer koda na slici 3.12.

```

network.update();
RF24NetworkHeader header01(nano01);
state01 = network.write(header01, &weatherTemperature,
sizeof(weatherTemperature));
RF24NetworkHeader header02(nano02);
state02 = network.write(header02, &weatherHumidity, sizeof(weatherHumidity));

```

*Sl. 3.12. Slanje podataka.*

### 3.2.5. Ispis podataka na OLED zaslon

Nakon slanja podataka, uspješnog ili neuspješnog, podatci o temperaturi i vlazi se svejedno ispisuju na OLED zaslon. Poziva se funkcija za ispisivanje vremenskih uvjeta na zaslonu, koja prima kao argumente temperaturu i vlagu (Slika 3.13). Pri ispisivanju uz argumente, na ekran se ispisuju naslovi koji označavaju što parametri predstavljaju, mjerne jedinice tih argumenata i oktalni zapis adrese gdje su ti argumenti poslani u mreži.

```

drawWeatherConditions(weatherTemperature, weatherHumidity);

```

*Sl. 3.13. Pozivanje funkcije za ispis na zaslon.*

U nastavku će biti opisani važniji dijelovi programskog kôda prijemne jedinice (Arduino Nano), a cijeli kôd se može pronaći u prilogu koji se nalazi na CD-u.

### 3.2.6. Primanje podataka putem NRF24 modula

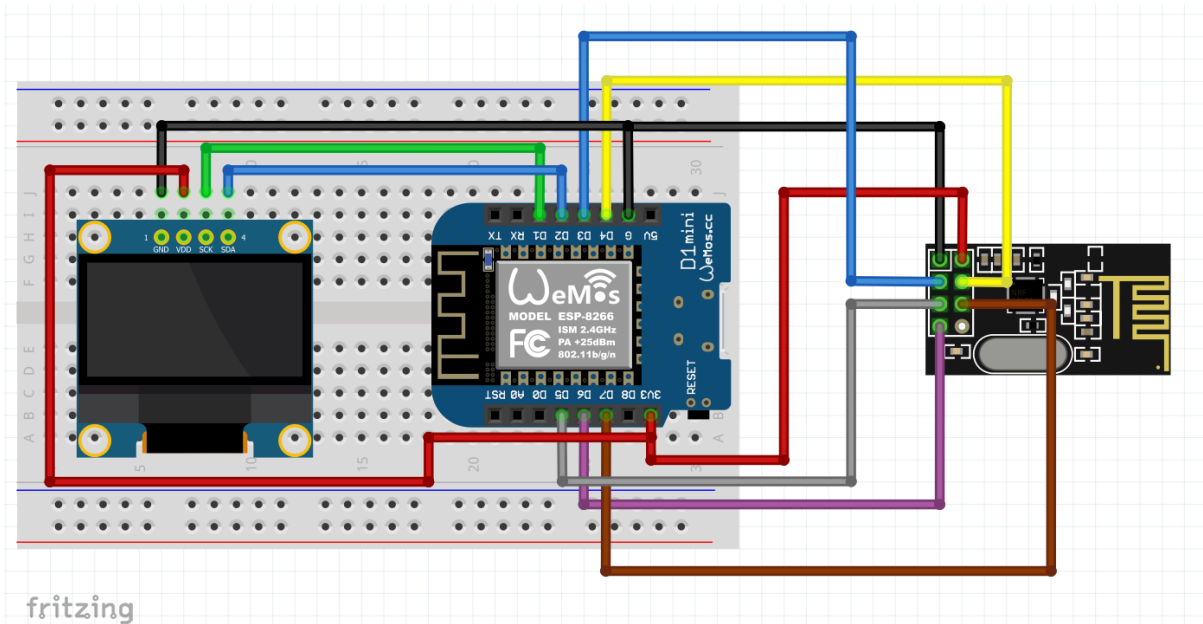
Ako je glavna jedinica uspješno poslala podatke na točnu adresu, prijamna jedinica će kod sljedećeg provjeravanja pristiglih podataka, te podatke spremiti u lokalnu varijablu, te odmah pozvati funkciju za ispisivanje tih podataka na OLED zaslonu (Slika 3.14). Ukoliko prođe 30 sekundi od prethodnog primitka podataka, prijamna jedinica poziva funkciju za ispis upozorenja o prekinutoj komunikaciji između nje i glavne jedinice.

```
network.update();  
while(network.available()){  
  lastCall = millis();  
  RF24NetworkHeader header1;  
  int weatherTemperature;  
  network.read(header1, &weatherTemperature, sizeof(weatherTemperature));  
  drawTemperature(weatherTemperature);  
}
```

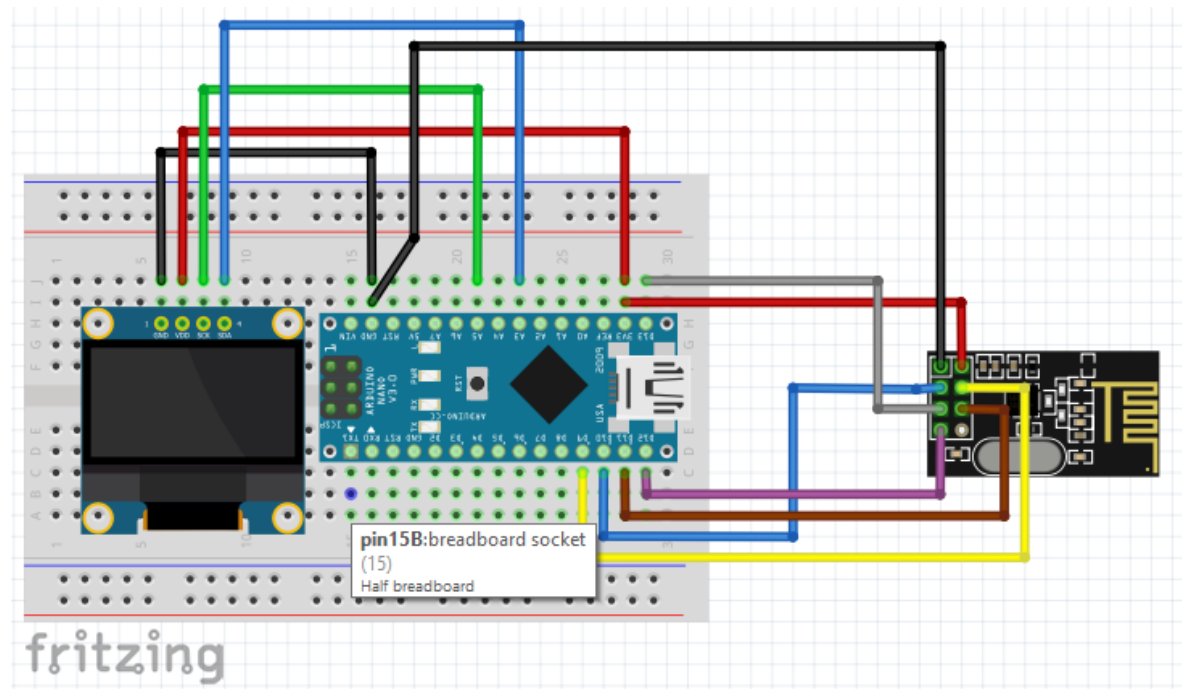
*Sl. 3.14. Provjeravanje pristiglih podataka.*

### 3.3. Razvijeni uređaji

Prije učitavanja napisanog kôda, na razvojnu pločicu je potrebno pravilno spojiti sve komponente koje se koriste u sustavu. Na slici 3.16 je prikazano pravilno spajanje komponenti glavne jedinice, te na slici 3.15 je prikazano pravilno spajanje za prijemne jedinice, koje su identične, a razlikuju se po kôdu koji se na njih učitava. Uz komponente prikazane na slikama, potrebno je imati 2 mini-B USB-a (za prijemne jedinice) i jedan micro USB kabel (za glavnu jedinicu) za napajanje razvojnih pločica.



Sl. 3.16. Shema spajanja glavne jedinice.

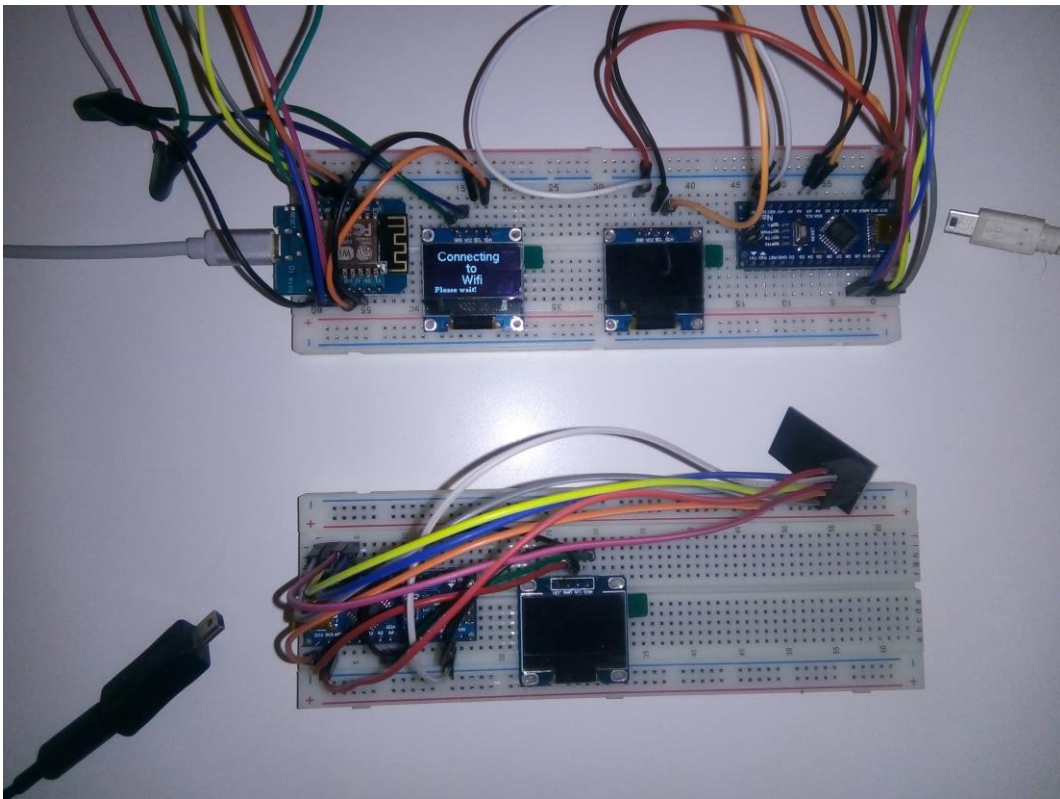


Sl. 3.15. Shema spajanja prijemnih jedinica.

### 3.4. Testiranje

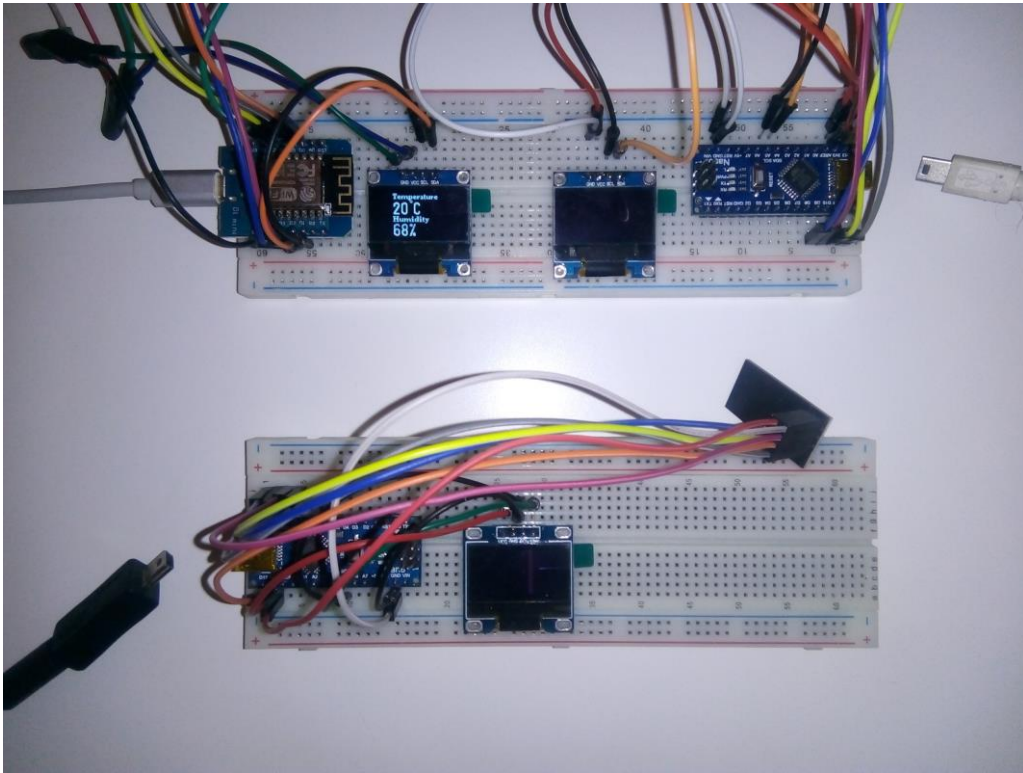
Ako su sve komponente ispravno povezane prema shemama u prethodnom dijelu, kôdovi učitani na odgovarajuće razvojne pločice, kroz sljedeći dio rada testirat će se kako one međusobno komuniciraju. Bitno je napomenuti da se u *header* datoteci na glavnoj jedinici treba napisati ispravno korisničko ime i lozinka Wifi mreže na koju se želimo spojiti

Ako su sve provjere obavljene glavna jedinica se može spojiti na izvor napajanja. Nakon priključivanja na izvor napajanja na zaslonu se prikazuje tekst „*Connecting to Wifi*“ prije negoli se modul spoji na internetsku mrežu što je vidljivo na slici 3.17.



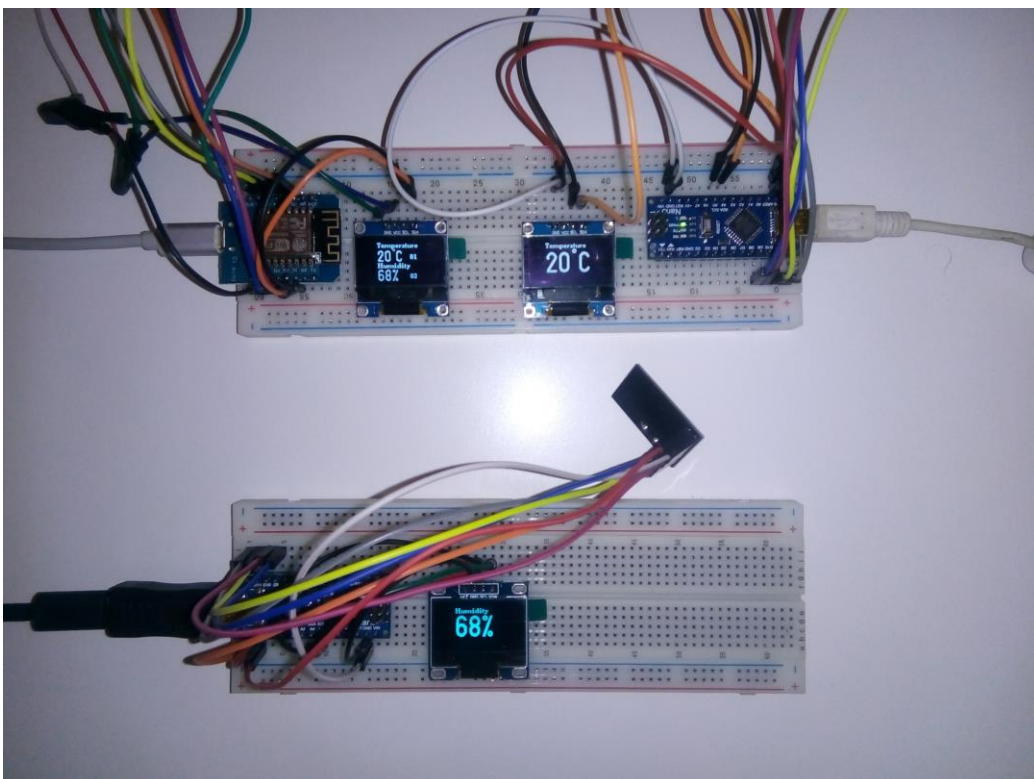
Sl. 3.17. Prikaz spajanje na Wifi.

Nakon par sekundi, ako je konekcija uspješna i ako je uspješno odrađen API poziv i parsiranje JSON objekta, na zaslonu se ispisuje trenutna temperatura (u stupnjevima celzijusa) i vlaga zraka (u postotcima) grada Osijeka prikazana na slici 3.19. Blinkanjem ledice na Wemos D1 mini razvojnoj pločici, može se uočiti da se svakih 10 sekundi te vrijednosti ažuriraju.



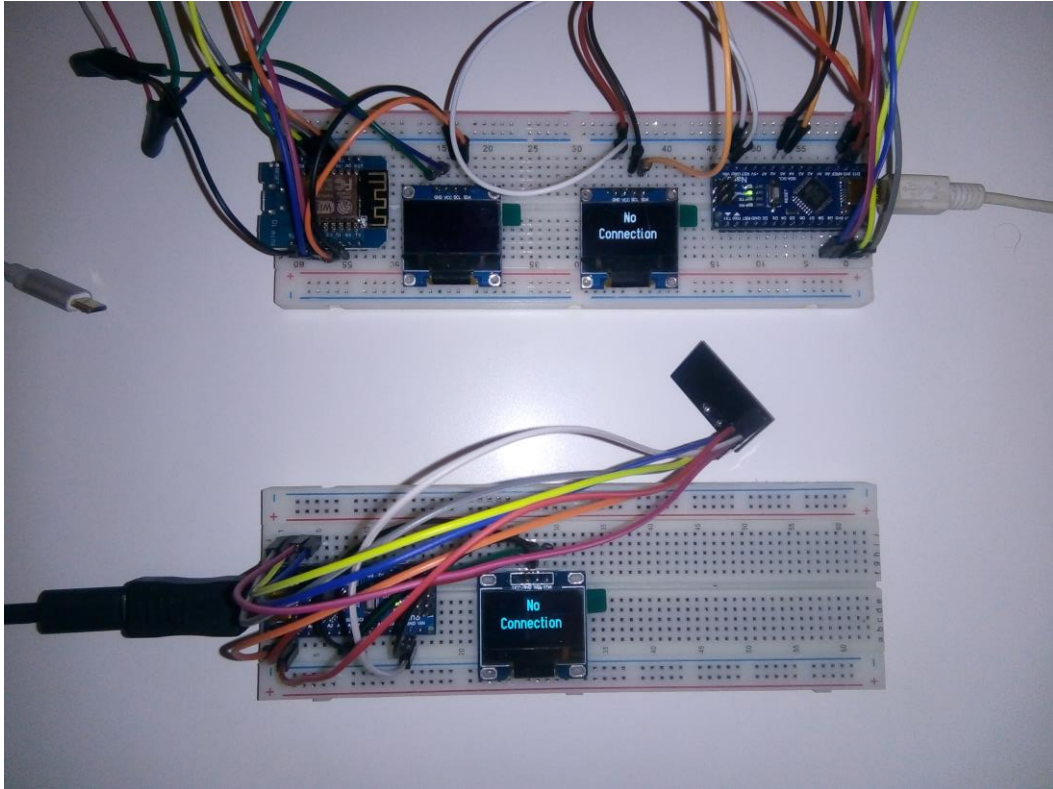
Sl. 3.19. Prikaz podataka prije slanja.

Nakon uspješnog spajanja glavne jedinice, mogu se priključiti druge dvije razvojne pločice na izvor napajanja. Može se uočiti da nakon sljedećeg ažuriranja glavne jedinice, jedna prijemna jedinica prikazuje istu temperaturu, a druga vlagu zraka identične onima na glavnoj jedinici. Isto se može uočiti da se na glavnoj jedinici pojavljuju adrese „01“ i „02“ pored svake vrijednosti prikazane na slici 3.18, koje ukazuju na uspješnost slanja podataka glavne jedinice prema prijemnima. Same vrijednosti označavaju oktalni zapis adrese prijemnih jedinica u mreži.



Sl. 3.18. Prikaz podataka nakon slanja.

Ako se isključi glavna jedinica i pričeka se neko vrijeme, prijemne jedinice ispisuju tekst „No Connection“ na zaslonu (vidljivo na slici 3.20), što ukazuje kako je prošlo 30 sekundi od zadnjeg primanja podataka s glavne jedinice.



*Sl. 3.20. Prikaz nakon isključivanja glavne jedinice.*

Iz priloženog se vidi kako je ostvarena dvosmjerna komunikaciju u mreži. Glavna jedinica šalje podatke prijemnima, dok one povratno šalju potvrdu o primljenim podacima. Komunikacija između NRF24 modula je testirana na otvorenom prostoru i dosiže udaljenosti do 100 m bez ikakvih smetnji.

## 4. ZAKLJUČAK

Bežična komunikacija u mreži sa NRF24 modulima uspješno je napravljena. Omogućava prikupljanje podataka o trenutačnim vremenskim uvjetima (temperatura i vlaga) za grad Osijek putem interneta na glavnoj jedinici. To ostvaruje putem API poziva, koja kao odgovor vraća JSON datoteku u kojoj se nalaze nama potrebne vrijednosti. Glavna jedinica ispisuje dobivene podatke na zaslon i prosljeđuje ih drugim prijemnim jedinicama u mreži, koje primljene podatke isto ispisuju na zaslon te povratno glavnoj jedinici šalju potvrdu o primljenim podacima. Svaka razvojna pločica u sklopu se napaja putem USB kabla na napon od 5 V. Podatci se ažuriraju i šalju svakih 10 sekundi na glavnoj jedinici.

Neki načini poboljšavanja sustava su dodavanje većeg broja prijemnih jedinica. Primjerice može se imati prijemna jedinica za prikaz vremenskih podataka u svakoj prostoriji kućanstva. Moguće je i povećati broj vremenskih parametara koji se očitavaju sa zaslona. JSON datoteka sadrži razne druge podatke osim temperature i vlage koje bi se mogle koristiti (tlak, maglovitost, oblačnost itd.). Sklopovlje se može nadograditi tako da radi s vanjskim izvorima napajanja kao što su baterije, te bi se time dobio prijenosni sustav, koji se može koristiti bilo gdje.

## LITERATURA

- [1] Tait Radio Academy: <https://www.taitradioacademy.com/topic/how-does-modulation-work-1-1/>. [Pristupljeno 22.8.2020]
- [2] RF Wireless World: <https://www.rfwirelessworld.com/Terminology/GFSK-vs-FSK.html>. [Pristupljeno 22.8.2020].
- [3] Last Minute Engineers: <https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/> [Pristupljeno 22.8.2020].
- [4] Tmrh20: <https://tmrh20.github.io/RF24Network/Tuning.html>. [Pristupljeno 22.8.2020].
- [5] How To Mechatronics: <https://howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules/>. [Pristupljeno 22.8.2020].
- [6] Stack Abuse: <https://stackabuse.com/what-is-arduino/>. [Pristupljeno 25.8.2020].
- [7] DiyIoT: <https://diyi0t.com/arduino-nano-tutorial/>. [Pristupljeno 26.8.2020].
- [8] DiyIoT: <https://diyi0t.com/what-is-the-esp8266-pinout-for-different-boards/>. [Pristupljeno 26.8.2020].
- [9] Instructables: <https://www.instructables.com/id/Monochrome-096-i2c-OLED-display-with-arduino-SSD13/>. [Pristupljeno 20.5.2020].
- [10] NETTIGO: <https://nettigo.eu/products/oled-display-0-96-i2c-128x64-blue>. [Pristupljeno 20.5.2020].
- [11] Acedemany: <http://fab.academany.org/2019/labs/vigyanashram/students/jayadip-sarode/Assignment14/images/2.png>. [Pristupljeno 28.8.2020].
- [12] Arduino: <https://www.arduino.cc/en/reference/SPI>. [Pokušaj pristupa 23.8.2020].
- [13] Circuitbasics: <https://www.circuitbasics.com/how-to-set-up-spi-communication-for-arduino/>. [Pristupljeno 24.5.2020].
- [14] How To Mechatronics: <https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/>. [Pristupljeno 24.8.2020].
- [15] Tizen: <https://docs.tizen.org/iot/guides/peripheral-io-api-i2c/>. [Pristupljeno 24.5.2020].



## SAŽETAK

Zadatak ovoga rada je bila ostvarivanje dvosmjerne komunikacije u mreži od tri sklopa baziranih na NRF24L01+ radio primo-predajniku. Glavna jedinica se sastoji od Wemos D1 mini razvojne pločice, na koju su spojeni NRF24L01+ modul i OLED zaslon. Sklop ima mogućnost preuzimanja vremenskih podataka (temperature i vlage) s interneta te ispis istih na OLED zaslon i slanje na druga dva sklopovlja putem NRF24L01+ modula. Jednom šalje podatke o temperaturi, a drugom o vlazi. Druga dva sklopovlja se sastoje od Arduino Nano razvojne pločice na koju su spojeni isto NRF24L01+ modul i OLED zaslon. Ova sklopovlja imaju mogućnost ispisivati na OLED zaslon primljene podatke od glavne jedinice te nazad slati potvrdu o uspješno primljenim podacima.

Ključne riječi: Arduino, Wemos D1 mini, NRF24L01+, OLED zaslon, bežična mreža

## **ABSTRACT**

The task of this project was to achieve two-way communication in a network of three circuits based on NRF24L01+ radio transceivers. The main unit consists of a Wemos D1 mini development board, to which the NRF24L01 + module and an OLED display are connected. The circuit has the task to download weather data (temperature and humidity) from the Internet and print them on the OLED screen and send to other circuits via NRF24L01 + module. It sends temperature data to one and humidity to the other. The other two circuits consist of an Arduino Nano development board to which NRF24L01+ module and an OLED display are connected. These circuits have ability to print on the OLED screen the data received from the main unit and send back confirmation of the successfully received data.

Key words: Arduino, Wemos D1 mini, NRF24L01+, OLED display, wireless network

## ŽIVOTOPIS

Marko Tandara rođen je 01.04.1996. godine u Slavonskom Brodu u Republici Hrvatskoj. Odrastao je u Đakovu gdje je pohađao osnovnu školu Josipa Antuna Čolnića, te ju je završio 2011. godine. Iste godine je upisao prirodoslovnu-matematičku gimnaziju „Antuna Gustava Matoša“ u Đakovu koju je završio 2015. godine. Nakon polaganja državne mature upisuje prvi odabir studija, elektrotehnički fakultet u Osijeku smjer Računarstvo (sada Fakultet elektrotehnike, računarstva i informacijskih tehnologija).

---

## **PRILOZI**

P1 Programski kôd za sva tri mikroupravljača se nalazi na CD-u