

# Kontrola kvalitete zraka korištenjem IoT platforme

---

**Delija, Goran**

**Master's thesis / Diplomski rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University North / Sveučilište Sjever**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:122:241977>

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

*Download date / Datum preuzimanja:* **2025-04-02**



*Repository / Repozitorij:*

[University North Digital Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

**SVEUČILIŠTE SJEVER  
SVEUČILIŠNI CENTAR VARAŽDIN**



DIPLOMSKI RAD br. 053/STR/2021

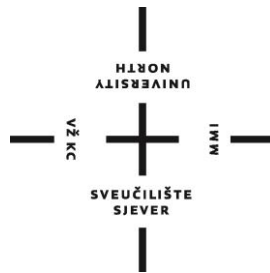
**Kontrola kvalitete zraka korištenjem IoT  
platforme**

Goran Delija

Varaždin, rujan 2021.



**SVEUČILIŠTE SJEVER**  
**SVEUČILIŠNI CENTAR VARAŽDIN**  
**Studij strojarstva**



DIPLOMSKI RAD br. 053/STR/2021

**Kontrola kvalitete zraka korištenjem IoT  
platforme**

Student:  
Goran Delija, 0908/336D

Mentor:  
prof. dr. sc. Marinko Stojkov

Varaždin, rujan 2021.

# Prijava diplomskog rada

## Definiranje teme diplomskog rada i povjerenstva

|                             |  |         |                                    |
|-----------------------------|--|---------|------------------------------------|
| ODJEL                       | Odjel za strojarstvo   |         |                                    |
| STUDIJ                      | diplomski sveučilišni studij Strojarstvo                     |         |                                    |
| PRISTUPNIK                  | Goran Delija   | IMBAG   | 0908/336D                          |
| DATUM                       | 29.09.2021.  | KOLEGIJ | Mehatronički sustavi               |
| NASLOV RADA                 | Kontrola kvalitete zraka korištenjem IoT platforme           |         |                                    |
| NASLOV RADA NA ENGL. JEZIKU | Air quality control using IoT platform                       |         |                                    |
| MENTOR                      | Prof.dr.sc. Marinko Stojkov                                  | ZVANJE  | Redoviti profesor u trajnom zvanju |
| ČLANOVI POVJERENSTVA        | 1. Izv. prof. dr. sc. Sanja Šolić, predsjednica povjerenstva |         |                                    |
|                             | 2. Prof. dr. sc. Marinko Stojkov, mentor, član               |         |                                    |
|                             | 3. doc. dr. sc. Matija Bušić, rezervni član                  |         |                                    |
|                             | 4. doc.dr.sc. Tomislav Veliki                                |         |                                    |
|                             | 5.   |         |                                    |

## Zadatak diplomskog rada

BROJ 053/STR/2021

OPIS

Pristupnik u svom diplomskom radu treba obraditi sljedeće:

- Opisati koncept "Interneta stvari" (IoT-a), područja primjene i strukturu IoT-a.
- Odabrati senzore i mikrokontroler za njihovo upravljanje, opisati njihove karakteristike i princip rada.
- Objasniti princip MQTT protokola za prijenos podataka.

- U praktičnom dijelu izraditi uređaj za kontrolu kvalitete zraka, sa prikupljanjem i obradom podataka sa senzora, te njihov prikaz u Web aplikaciji. Konfigurirati odabrano računalo kao MQTT broker, te napraviti Web aplikaciju za prikaz izmjerenih veličina sa senzora. Navesti izrađene programske kôdove i njihova objašnjenja.

ZADATAK URUČEN

1. 10. 2021.



PODPIS MENTORA

M. Stojkov

# **Predgovor**

Izjavljujem da sam ovaj rad izradio samostalno, koristeći navedenu literaturu i znanja stečena tijekom studija.

Zahvaljujem se prof. dr. sc. Marinku Stojkovu na prihvaćanju mentorstva, pomoći, savjetima te utrošenom vremenu i trudu. Također zahvaljujem svim profesorima Sveučilišta Sjever na njihovim savjetima i prenesenom znanju tijekom studija.

Posebna zahvala ide mojim roditeljima na pruženoj potpori i strpljenju za vrijeme cijelog studiranja.

## Sažetak

U diplomskom radu opisan je koncept „Interneta stvari“ ili IoT (eng. *Internet of Things*) platforme kroz izradu uređaja za mjerenje kvalitete zraka sa prikupljanjem i obradom podataka u stvarnom vremenu, te prikazom u Web aplikaciji. Navedeni uređaj sastoji se od mikrokontrolerske pločice Wemos D1 Mini kao središnjom jedinicom, baziranoj na čipu ESP8266. Wemos D1 prikuplja podatke sa senzora CCS811 i BME280. CCS811 je senzor kvalitete zraka koji mjeri količinu raznih čestica u zraku, a BME280 je atmosferski senzor koji mjeri tri veličine: temperaturu, tlak i vlažnost zraka. Povezivanje s MQTT brokerom vrši se bežično Wi-Fi mrežom, koristeći MQTT protokol. Objasnjeno je koncept IoT-a, njegova važnost i primjena u industriji. Opisan je Wemos D1 Mini, korišteni senzori te su navedene njihove karakteristike. Ukratko su navedene karakteristike Raspberry Pi mini računala. Objasnjeno je MQTT protokol i način njegovog korištenja. Opisan je postupak programiranja Wemos D1, Raspberry Pi-ja, konfiguriranja MQTT brokera.

**Ključne riječi:** IoT, Internet stvari, Wemos D1 Mini, Raspberry Pi, CCS811, BME280, kontrola kvalitete zraka, web aplikacija, MQTT

## **Abstract**

In this master's thesis, the concept of "Internet of Things" or IoT platform is explained through the making of device which measures the air quality, collects and processes data in real time, and displays it in a Web application. This device consists of a microcontroller board Wemos D1 Mini as it's central unit, which is based on ESP8266 microchip. Wemos D1 Mini collects data from CCS811 and BME280 sensors. CCS811 is an air quality sensor which measures the quantity of particles in the air, and BME280 is an atmospheric sensor which measures three sizes: temperature, pressure and relative humidity. Connecting with the MQTT broker is made wireless via Wi-Fi network, using the MQTT protocol. The concept of IoT is explained, its importance and usage in industry. Wemos D1 Mini is described, as are the used sensors. The characteristics of Raspberry Pi mini computer are briefly stated. The functioning of the MQTT protocol and its usage are explained. The process of programming is explained for Wemos D1, Raspberry Pi, as is the process of configuring the MQTT broker.

**Keywords:** IoT, Internet of Things, Wemos D1 Mini, Raspberry Pi, CCS811, BME280, air quality control, web application, MQTT



## Popis korištenih kratica

|                        |   |
|------------------------|---|
| <b>eng.</b>            | engleski  |
| <b>tj.</b>             | to jest   |
| <b>itd.</b>            | i tako dalje  |
| <b>dr.</b>             | drugo   |
| <b>npr.</b>            | na primjer  |
| <b>IoT</b>             | Internet of Things<br>Internet stvari   |
| <b>MQTT</b>            | Message Queuing Telemetry Transport<br>mrežni protokol za komunikaciju između dva uređaja |
| <b>Wi-Fi</b>           | tehnologija bežičnog lokalnog umrežavanja   |
| <b>M2M</b>             | Machine to Machine<br>komunikacija između uređaja   |
| <b>RFID</b>            | Radio – frequency identification<br>identifikacija pomoću radio-frekvencije               |
| <b>kbps</b>            | kilobits per second<br>kilobita po sekundi  |
| <b>Mbps</b>            | megabits per secons<br>megabita po sekundi  |
| <b>kHz</b>             | kiloherc  |
| <b>NB-IoT</b>          | Narrowband Internet of Things<br>uskopojasni Internet stvari                              |
| <b>BLE</b>             | Bluetooth Low Energy<br>Bluetooth niske potrošnje   |
| <b>LoRa</b>            | Long Range<br>modulacijska tehnologija velikog dometa                                     |
| <b>LoRaWAN</b>         | Long Range Wide Area Network<br>mreža velikog raspona širokog dometa                      |
| <b>TVOC</b>            | Total Volatile Organic Compounds<br>ukupno isparljivi organski spojevi                    |
| <b>eCO<sub>2</sub></b> | ekvivalent ugljičnog dioksida   |
| <b>MOX</b>             | metalni oksidi  |
| <b>I<sub>2</sub>C</b>  | Inter – integrated circuit  |

|             |  |
|-------------|--|
|             | vrsta serijske komunikacijske sabirnice                                      |
| <b>SCL</b>  | serial clock<br>linija za prijenos takta kod I <sub>2</sub> C komunikacije   |
| <b>SDA</b>  | serial data<br>linija za prijenos podataka kod I <sub>2</sub> C komunikacije |
| <b>V</b>    | volt   |
| <b>mA</b>   | miliamper  |
| <b>mW</b>   | miliwat  |
| <b>m</b>    | metar  |
| <b>hPa</b>  | hektopaskal  |
| <b>MB</b>   | megabajt   |
| <b>GPIO</b> | General Input Output Pin<br>standardni ulazno – izlazni pinovi               |
| <b>QoS</b>  | Quality of service<br>kvaliteta usluge kod MQTT protokola                    |
| <b>ppm</b>  | parts per million<br>dijelova na milijun                                     |
| <b>ppb</b>  | parts per billion<br>dijelova na bilijun                                     |

# Sadržaj

|      |  |    |
|------|--|----|
| 1.   | Uvod.....  | 1  |
| 2.   | Općenito o IoT-u (Internetu stvari).....               | 2  |
| 2.1. | Povijest IoT-a .....                                   | 3  |
| 2.2. | Važnost IoT-a.....                                     | 3  |
| 3.   | Struktura IoT-a.....                                   | 5  |
| 3.1. | „Stvari“, senzori i aktuatori.....                     | 5  |
| 3.2. | Prikupljanje podataka i pristupne točke .....          | 6  |
| 3.3. | Rubni računalni sustavi .....                          | 7  |
| 3.4. | Podatkovni centri, sustavi smješteni na „oblaku“ ..... | 8  |
| 4.   | Pregled bežičnih IoT tehnologija .....                 | 9  |
| 4.1. | Wi-Fi .....  | 9  |
| 4.2. | NB-IoT .....   | 10 |
| 4.3. | BLE .....  | 10 |
| 4.4. | LoRa .....   | 11 |
| 5.   | Komponente sustava za kontrolu kvalitete zraka.....    | 13 |
| 5.1. | CCS811 senzor kvalitete zraka.....                     | 13 |
| 5.2. | BME280 senzor.....                                     | 16 |
| 5.3. | ESP8266.....   | 18 |
| 5.4. | Raspberry Pi .....                                     | 21 |
| 6.   | MQTT protokol.....                                     | 24 |
| 6.1. | MQTT broker .....                                      | 25 |
| 6.2. | MQTT klijent .....                                     | 25 |
| 6.3. | Razina kvalitete usluge kod MQTT protokola.....        | 25 |
| 7.   | Programiranje Raspberry Pi mini računala .....         | 26 |
| 7.1. | Konfiguriranje MQTT brokera.....                       | 26 |
| 7.2. | Python aplikacija za prikaz rezultata.....             | 27 |
| 8.   | Programiranje Wemos D1 Mini .....                      | 29 |
| 9.   | Mjerenje kvalitete zraka u realnim uvjetima .....      | 36 |
| 10.  | Zaključak.....   | 39 |
| 11.  | Literatura.....  | 41 |
| 12.  | Popis slika .....                                      | 43 |
| 13.  | Popis tablica .....                                    | 44 |
| 14.  | Popis grafova .....                                    | 45 |

# 1. Uvod

Zadnjih nekoliko godina, u medijima se često može čuti pojam „Internet stvari“. Uređaje za koje je do nedavno bilo nezamislivo da budu spojeni na Internet, poput kućanskih aparata i automobila, njihovi proizvođači danas reklamiraju kao „pametne“, te je teško pronaći proizvod koji nema IoT verziju. Ogromna pokrivenost brzim mobilnim mrežama, kao i cijena elektroničkih komponenti koja se smanjila na vrlo nisku razinu, omogućili su entuzijastima i hobistima da uz vrlo mali trošak i sami naprave svoje verzije takvih pametnih uređaja.

Osnovna ideja za ovaj rad bila je izraditi IoT senzor koji će moći raditi na baterije pa neće imati ograničenja za radno mjesto postavljanja, te koji će bežično slati očitane vrijednosti na uređaj koji je spojen sa Internetom. Korisnicima je tada omogućeno da vrijednosti sa senzora mogu u svakom trenutku provjeriti korištenjem Web aplikacije, npr. pomoću svojeg pametnog mobitela.

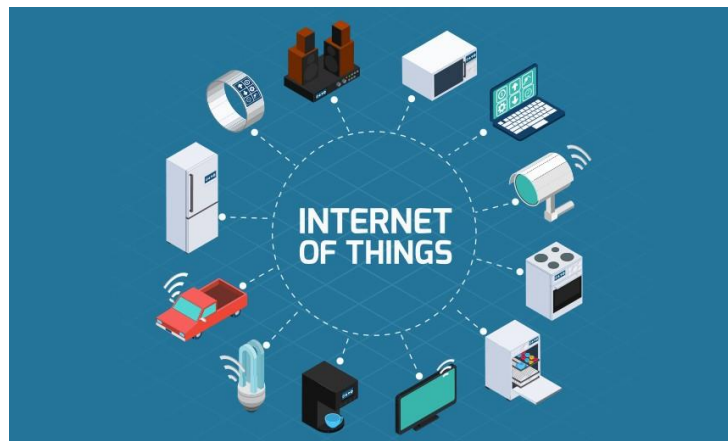
Odabrana je izrada senzora za kontrolu kvalitete zraka, koji mjeri količinu raznih čestica u zraku. Takav senzor mogao bi imati široku primjenu u strojarstvu: poduzeća koja se bave npr. obradom metala mogle bi u svakom trenutku imati uvid u kakvoću zraka u svojim proizvodnim halama, kako bi bile sigurne u sigurnost svojih zaposlenika. U slučaju da neki proces onečisti zrak iznad neke preporučene razine, sustav bi mogao obavijestiti odgovornu osobu te bi ona tada mogla poduzeti neku radnju, npr. provjetriti prostorije ili udaljiti radnike iz njih.

Kao glavna jedinica senzora odabrana je razvojna pločica bazirana na ESP8266 mikrokontroleru, Wemos D1 Mini, na koju su putem I<sup>2</sup>C sabirnice spojeni CCS811 senzor kvalitete zraka i BME280 atmosferski senzor. Za primanje podataka sa razvojne pločice odabrano je Raspberry Pi 3B mini računalo. Raspberry Pi je unazad nekoliko godina napravio pravu revoluciju jer sa svojim malim dimenzijama (veličine je kreditne kartice) u sebi sadrži sve dijelove kao i obično osobno računalo, naravno mnogo slabije, ali i dalje dovoljne snage za pokretanje i obavljanje mnogih zadataka koje se tiču korištenja mikrokontrolera i senzora.

## 2. Općenito o IoT-u (Internetu stvari)

IoT ili Internet stvari (eng. *Internet of Things*) opisuje globalnu mrežu fizičkih uređaja – „stvari“ – koja sadrži senzore, upravljačke programe i ostale tehnologije s ciljem povezivanja i dijeljenjem podataka sa drugim uređajima i sustavima putem Interneta. Raspon uređaja koji čine Internet stvari veoma je širok, od uobičajenih kućanskih aparata poput obične žarulje, hladnjaka i televizora, do specijaliziranih industrijskih alata.

Unazad nekoliko godina, Internet stvari postao je jedan od najvažnijih tehnologija dvadeset prvog stoljeća, te broj povezanih uređaja raste iz dana u dan. Razlog tome sve je bolja pokrivenost mobilnim, 4G i u zadnje vrijeme 5G mrežama, te drastičan pad cijena mikrokontrolera, senzora i modula za bežično povezivanje putem Wi-Fi mreža. Oni omogućuju pretvaranje skoro svakog uređaja u dio IoT-a, dajući mu pri tome neku vrstu digitalne inteligencije i mogućnost automatske razmjene podataka između dva uređaja, takozvana M2M (eng. *Machine to Machine*) komunikacija, bez potrebe za ljudskim uplitanjem. <sup>[1]</sup>



Slika 1 Širok raspon uređaja spojenih na IoT <sup>[2]</sup>

## 2.1. Povijest IoT-a

Glavni koncept mreže pametnih uređaja raspravljan je već 1982. godine, kada je modificirani Coca-Cola automat na sveučilištu Carnegie Mellon postao prvi uređaj spojen na Internet, sa mogućnošću javljanja svog sadržaja, kao i podatka jesu li nedavno skladišteni proizvodi hladni ili ne. Pojam Interneta stvari osmislio je Kevin Ashton iz tvrtke Procter & Gamble, 1999. godine, iako je on izvorno preferirao izraz „Internet *za* stvari“. U to vrijeme, smatrao je da je identifikacija pomoću radio-frekvencija (RFID) ključna tehnologija za razvoj IoT-a. Cisco sustavi definiraju Internet stvari kao trenutak u vremenu kada je broj uređaja spojenih na Internet premašio broj ljudi koji koriste Internet, te je po njihovoj procjeni IoT nastao negdje između 2008. i 2009. godine.

Internet stvari je u početku bio najzanimljiviji i najkorisniji velikim tvrtkama i proizvođačima, gdje se prvenstveno koristio za komunikaciju između uređaja (M2M) i za praćenje skupe opreme i alata u proizvodnji pomoću RFID oznaka. U zadnje se vrijeme fokus sve više prebacuje na transformiranje naših domova i ureda u „pametne prostore“. <sup>[3] [4]</sup>

## 2.2. Važnost IoT-a

Kao što je već rečeno, Internet stvari važna je tehnologija koja se svakim danom sve više usavršava. Mnogo je prednosti u tome što možemo međusobno povezati razne uređaje, a najvažnije su:

- **Više podataka znači bolje donošenje odluka** – Sa dodanim senzorima, takvi uređaji mogu skupljati velike količine podataka iz raznih područja. Npr., uz praćenje kojoj hrani se bliži istek roka trajanja u pametnom hladnjaku, takav uređaj može dati informacije o svojoj potrošnji električne energije, temperaturi, prosječnom vremenu otvaranja vrata i mnoge druge. Velik broj prikupljenih informacija znači da tvrtka koja proizvodi uređaj može analizirati te podatke sa svrhom poboljšanja karakteristika samog uređaja.
- **Mogućnost praćenja podataka** – Kao što praćenje podataka pomaže samom proizvođaču, tako pomaže i krajnjem korisniku. Praćenjem informacija o uređaju, njegovim radnim svojstvima, može se ustanoviti nepravilan rad i mogući kvar prije samog otkazivanja uređaja te obaviti njegova pravovremena zamjena.
- **Olakšanje poslova pomoću automatizacije** – Uređaj većinu posla obavlja automatski, bez ljudskog djelovanja.

- **Povećanje učinkovitosti i smanjenje troškova i resursa** – Uz uštedu samog vremena za korisnika, IoT može dovesti i do smanjenja troškova. Npr., automatsko isključivanje rasvjete i ostalih kućanskih uređaja u trenutku napuštanja prostorije ili kuće može pridonijeti smanjenju potrošnje električne energije.
- **Bolja kvaliteta života** – Na kraju, sve nabrojane prednosti pridonose poboljšanju kvalitete života. Uređaj koji obavlja trivijalne stvari poput isključenja svjetala i naručivanja namirnica, ostavlja korisniku više vremena da se bavi važnijim stvarima te pridonosi smanjenju stresa. [5]

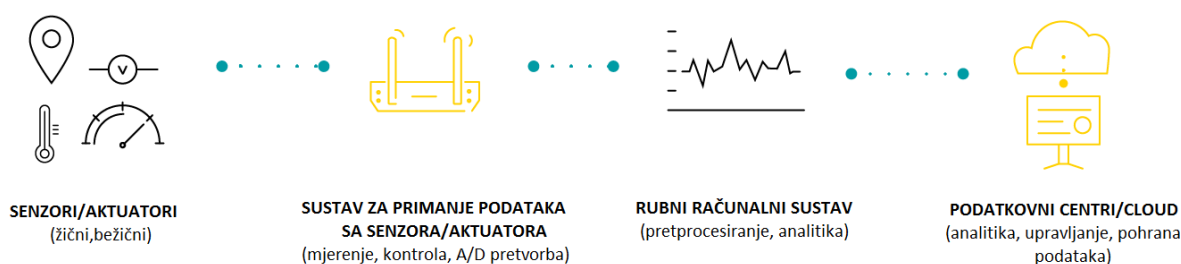


Slika 2 Automatizacija doma pomoću IoT-a sve je češća [6]

### 3. Struktura IoT-a

Iako je svaki IoT sustav različit, temeljna arhitektura, kao i proces prijenosa podataka, sličan je za svaki takav sustav. Prvi sloj sastoji se od „stvari“. Njega čine objekti spojeni na Internet koji pomoću ugrađenih senzora i aktuatora mogu detektirati određene vrijednosti iz okoline i zatim te veličine proslijediti IoT pristupnim točkama. Drugi sloj sastoji se od pristupnih točaka za primanje podataka sa senzora ili aktuatora, koji primaju neobrađene podatke, pretvaraju ih u digitalni oblik te ih filtriraju tako da su spremni za daljnju analizu. Treći sloj čine rubni računalni sustavi zaduženi za pretprocesiranje i analizu podataka. U ovom sloju također se može odraditi vizualizacija podataka i strojno učenje na temelju dobivenih vrijednosti. U zadnjem, četvrtom sloju, podaci se šalju u podatkovne centre koji mogu biti instalirani lokalno ili bazirani u oblaku (eng. *cloud based*). U njima se podaci spremaju, vrši se upravljanje te se rade složenije analize. U nastavku poglavlja detaljnije će se opisati svaki pojedini sloj.

Sljedeća slika prikazuje četiri sloja IoT arhitekture.



Slika 3 Četiri sloja IoT arhitekture <sup>[7]</sup>

#### 3.1. „Stvari“, senzori i aktuatori

Srž svakog IoT sustava su uređaji koji služe za prikupljanje podataka. Da bi imali mogućnost prikupljanja podataka iz samog objekta ili vanjskog svijeta, potrebni su im senzori. Oni mogu biti ugrađeni kao komponente u drugim uređajima ili mogu biti samostalni vanjski moduli koji mjere i šalju očitane vrijednosti. Npr., zadaća agrikulturnog senzora može biti mjerenje parametara poput temperature tla i zraka, vlažnosti, pH vrijednosti tla ili izloženosti usjeva Sunčevim zrakama. Drugi dio prvog sloja čine aktuatori. U bliskoj suradnji sa sensorima, oni mogu pretvoriti podatke dobivene od pametnih uređaja u fizičku radnju. Tako pametan sustav za navodnjavanje sa svim sensorima može, na temelju podataka dobivenih sa tih senzora, analizirati situaciju u stvarnom vremenu i pomoću aktuatora otvoriti odabrane ventile na mjestima gdje je vlažnost tla niža od



minimalne zadane vrijednosti. Ventili će ostati otvoreni sve dok senzori ne jave da se vrijednost vlažnosti vratila na zadanu. Naravno, sve ovo se odvija bez ljudskog utjecaja.

Važna značajka ovih sustava je i mogućnost komunikacije uređaja međusobno, odnosno njihovo prikupljanje i dijeljenje podataka. U slučaju sustava sa ograničenim resursima i baterijskog napajanja, postizanje toga nije nimalo jednostavan posao budući da takva komunikacija zahtijeva dosta računalnih resursa koji su zahtjevni za napajanje te ovise o ionako ograničenoj propusnosti podataka. Zbog toga, robusna arhitektura može omogućiti efikasno upravljanje uređajem samo korištenjem modernih, sigurnih i kompaktnih komunikacijskih protokola, poput M2M komunikacije.

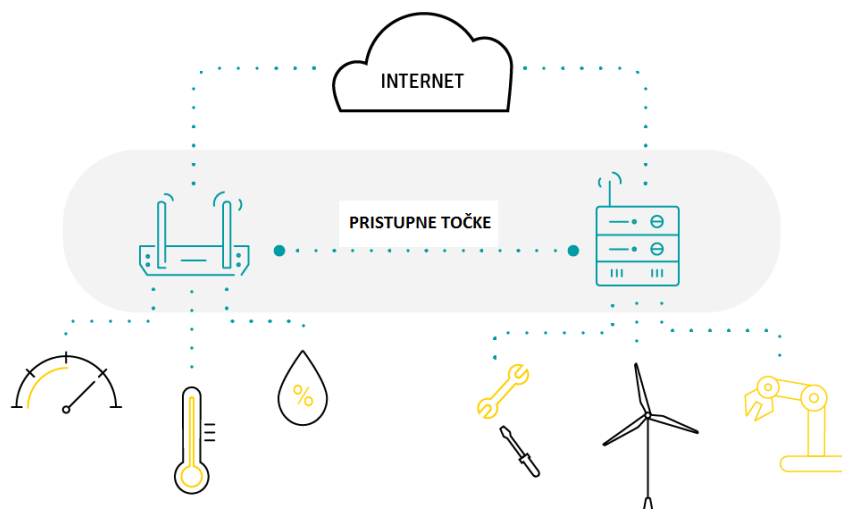


Slika 4 Prikaz prvog sloja IoT arhitekture [7]

### 3.2. Prikupljanje podataka i pristupne točke

Iako ovaj sloj radi u blizini senzora i aktuatora na danim uređajima, važno ga je opisati kao zaseban sloj IoT arhitekture budući da je važan za obradu prikupljenih podataka, njegovo filtriranje i slanje rubnoj računalnoj infrastrukturi i platformama baziranih u oblaku. Senzori šalju ogromne količine podataka pa je njihovo pravilno filtriranje i slanje od iznimne važnosti. Kao posrednici između senzora i sustava za analitičku obradu, pristupne točke (eng. *gateways*) omogućuju komunikaciju između slojeva tako što pretvaraju podatke sa senzora u formate koji se lako prenose i koriste u daljnjim slojevima. Kontroliraju, filtriraju i odabiru podatke koji će se slati kako bi se smanjila količina prenesenih podataka, što ima za rezultat manje troškove mrežne komunikacije i niže vrijeme potrebno za odgovor od strane sustava za upravljanje. Pristupne točke tako vrše početno, lokalno pretprocesiranje podataka sa senzora, organizirajući ih u pakete spremne za daljnju obradu.

Još jedna zadaća pristupnih točaka je i sigurnost. Budući da su zaduženi za upravljanjem toka informacija u oba smjera, pomoću odgovarajućih enkripcija i sigurnosnih alata mogu spriječiti „curenje“ IoT podataka smještenih u oblaku, kao i smanjiti rizik od zlonamjernih vanjskih napada na IoT uređaje.

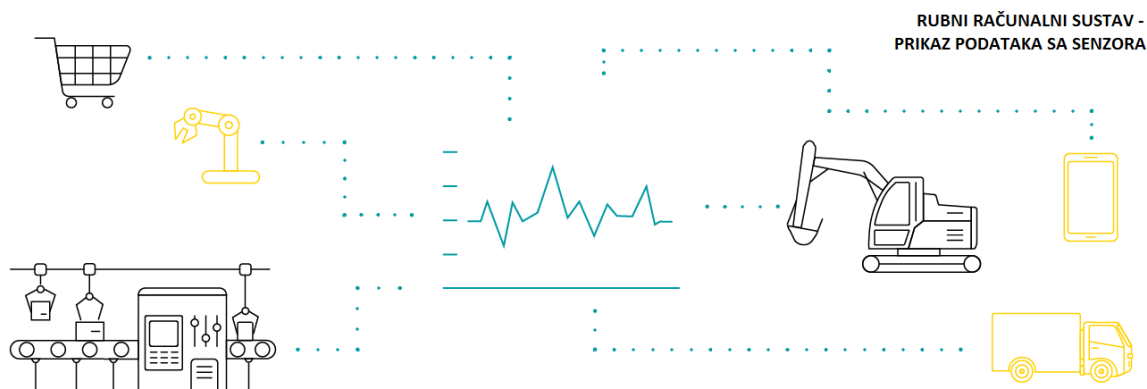


Slika 5 Prikaz drugog sloja IoT arhitekture <sup>[7]</sup>

### 3.3. Rubni računalni sustavi

Iako nisu neizostavni dio svake IoT arhitekture, rubni računalni sustavi mogu pridonijeti poboljšanju IoT projekata, posebice onih širokog obujma. Zbog moguće ograničene dostupnosti i brzine prijenosa podataka prema mrežnim IoT platformama, takvi sustavi mogu smanjiti vrijeme odaziva upravljačkih sustava i poboljšati fleksibilnost analitičke obrade podataka. Kako je brzina obrade podataka ključna u nekim primjenama Interneta stvari, rubni računalni sustavi su u proteklim godinama doživjeli sve veći rast popularnosti u industrijskim aplikacijama IoT-a.

Budući da su fizički smješteni bliže izvoru podataka, jednostavnije je i lakše takvom sustavu osigurati neku vrstu povratne informacije. U tom slučaju, samo se veći i opsežniji paketi podataka, koji su kompliciraniji za obradu, šalju dalje na zadnji, četvrti sloj. Smanjenjem podataka koji se mrežno šalju, može se unaprijediti sigurnost sustava, kao i smanjiti njegova potrošnja. Primjer ovog sloja može biti mobilna aplikacija koja prikazuje osnovne podatke senzora, dok se složenija analiza tih podataka, kao i njihovo skladištenje odvija na nekom serveru ili sustavu smještenom na oblaku.



Slika 6 Prikaz trećeg sloja IoT arhitekture [7]

### 3.4. Podatkovni centri, sustavi smješteni na „oblaku“

Podatkovni centri čine „mozak“ IoT sustava. Za razliku od rubnih računalnih sustava, podatkovni centri ili sustavi smješteni na oblaku dizajnirani su za spremanje, obradu i analizu masivnih količina podataka kako bi se dobilo dublje razumijevanje očitanih vrijednosti. Oni koriste napredne sustave za analizu podataka (eng. *data analytics engines*) i mehanizme strojnog učenja koje rubni računalni sustavi ne podržavaju.

Industrijske IoT arhitekture, koje su proteklih godina počele koristiti najbolje podatkovne centre za obradu podataka, kao rezultat su dobile poboljšanja u vidu povećane produktivnosti proizvodnje, smanjenje neplaniranog zastoja u proizvodnji kao rezultat kvarova te smanjenje potrošnje električne energije. [7]

Zajedno s pravilnim korištenjem od strane krajnjih korisnika, podatkovni centri mogu pružiti poslovnu inteligenciju i opcije vizualizacije rezultata koje olakšavaju rad zaposlenicima na njihovom radnom mjestu, olakšavaju kontrolu i nadzor koji za rezultat imaju bolje donošenje odluka na temelju izvješća i praćenja podataka u stvarnom vremenu.



Slika 7 Prikaz četvrtog sloja IoT arhitekture [7]

## 4. Pregled bežičnih IoT tehnologija

S razvojem IoT industrije u proteklih nekoliko godina, pojavili su se i neki novi oblici bežične komunikacije između uređaja. Neki od protokola za bežičnu komunikaciju nastali su na temelju bežičnih tehnologija koje su se koristile mnogo godina prije nego je IoT kao koncept uopće osmišljen, kao na primjer BLE koji je proizašao iz Bluetooth Classic tehnologije, kojim su se prvenstveno povezivali audio uređaji sa računalima ili mobitelima.

Odabir protokola ovisi o namjeni našeg IoT projekta. Osnovni kriteriji za odabir su potrošnja, domet i propusnost (eng. *bandwidth*). Budući da IoT uređaji često nemaju mogućnost napajanja pomoću električne mreže, koriste baterije. U tom slučaju, važno je da je potrošnja uređaja što manja, kako bi njegova autonomija i vijek trajanja bili što veći. Često se kao zahtjev za mnoge IoT uređaje navodi autonomija od nekoliko godina. Što se dometa tiče, on može biti od nekoliko metara unutar zgrade (npr. Wi-Fi, BLE) do nekoliko kilometara na otvorenom prostoru (npr. LoRa). Najčešće se sa povećanjem dometa smanjuje propusnost, odnosno količina podataka koje IoT uređaj šalje, kako bi potrošnja ostala što manja.

Kratak opis najčešće korištenih bežičnih IoT protokola dan je u nastavku.

### 4.1. Wi-Fi

Wi-Fi povezivanje često je očiti izbor za mnoge IoT projekte, posebno s obzirom na ogromnu raširenost Wi-Fi signala, bilo u kućnim ili poslovnim okruženjima. Široko rasprostranjena postojeća infrastruktura nudi brz prijenos podataka i mogućnost rukovanja velikim količinama podataka.

Trenutno je najčešći Wi-Fi standard koji se koristi u domovima i mnogim tvrtkama 802.11n, koji nudi ogromnu propusnost u rasponu od stotina megabita u sekundi, što je odlično za prijenos velikih datoteka, ali može uzrokovati preveliko trošenje energije za mnoge IoT aplikacije. Uz to, Wi-Fi zahtijeva router preko kojeg su povezani svi uređaji u lokalnoj mreži i koji je zadužen za prijenos paketa između njih, te zbog toga on mora biti u dometu svih IoT uređaja.

Radi na frekvencijama od 2,4 GHz i 5 GHz, a domet mu je do 50 metara. Brzine prijenosa podataka su maksimalno 600 Mbps, češće 150-200 Mbps, ovisno o frekvenciji kanala i broju antena (najnoviji standard 802,11-ac trebao bi ponuditi 500 Mbps do 1 Gbps).<sup>[8]</sup>

## 4.2. NB-IoT

Uskopojasni Internet stvari (eng. *NB-IoT, Narrowband Internet of Things*), standard je radio tehnologije, širokopojasne mreže niske snage (eng. *LPWAN, Low Power Wide Area Network*), kojeg je razvio 3GPP, konzorcij zadužen za mrežne standarde, za upotrebu u mobilnim uređajima i uslugama.

NB-IoT posebno se usredotočuje na unutarnju pokrivenost, niske cijene, dugo trajanje baterije i veliku propusnost veze. NB-IoT koristi podskup LTE standarda, ali ograničava propusnost na jedan uskopojasni kanal od 200 kHz. Koristi OFDM modulaciju za downlink komunikaciju i SC-FDMA za uplink komunikaciju. NB-IoT je najbolje primjeren za IoT aplikacije koje zahtijevaju češću komunikaciju i prijenos podataka, zbog toga što djeluje na licenciranom frekvencijskom spektru te nema ograničenja radnog ciklusa. Maksimalna brzina downlinka je 26 kbps, a uplinka 62 kbps. <sup>[9][10]</sup>

U ožujku 2019., Globalna udruga mobilnih dobavljača (GSA) objavila je da više od 100 mobilnih operatora ima mreže NB-IoT ili LTE-M. Taj se broj do rujna 2019. popeo na 142 razvijene/pokrenute mreže. <sup>[10]</sup> Među njima su i HT i A1.



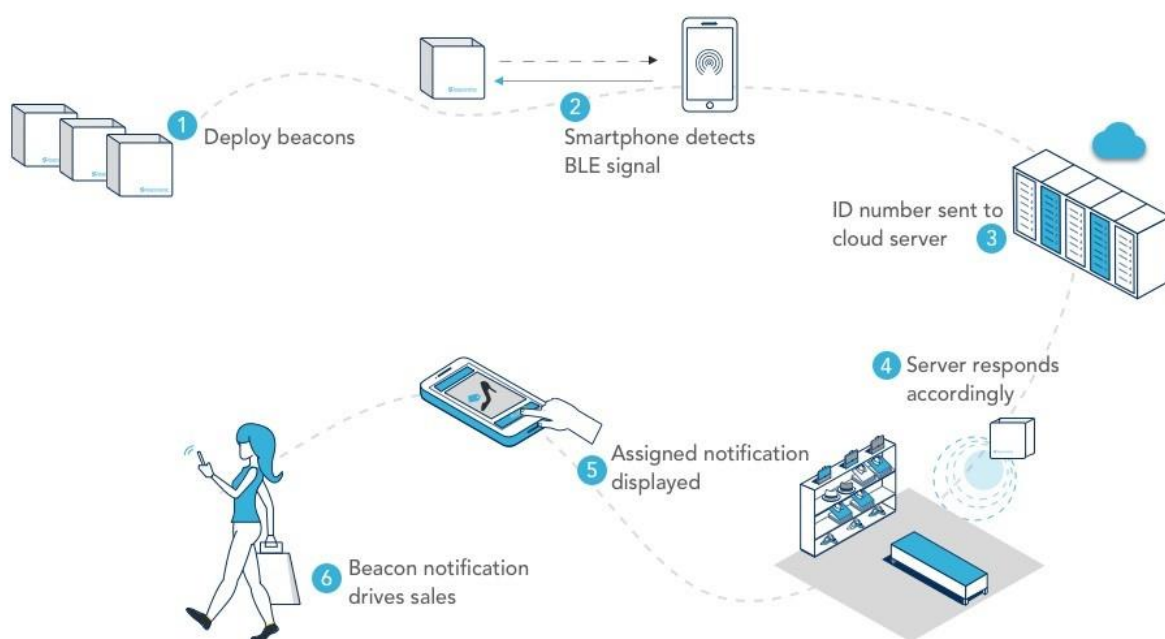
Slika 8 NB-IoT parkirni senzor varaždinske tvrtke Mobilis <sup>[11]</sup>

## 4.3. BLE

BLE (eng. *Bluetooth Low Energy – Bluetooth niske potrošnje*) bežična je tehnologija osmišljena i plasirana od strane Bluetooth SIG -a (eng. Bluetooth Special Interest Groupa, organizacije koja razvija i upravlja Bluetooth standardom). Prvenstveno je zamišljena kao nadogradnja Bluetooth Classic tehnologije za primjenu u IoT uređajima. Većinom se koristi u uređajima u zdravstvenom sektoru (stalna kontrola i mjerenje, npr, otkucaja srca, razine glukoze, krvnog tlaka), fitnessu, odašiljačima (eng. *beacon*), sigurnosti i industriji kućne zabave. <sup>[12]</sup>

U usporedbi s klasičnim Bluetoothom, glavne karakteristike Bluetooth Low Energy-ja su značajno smanjena potrošnja energije i niži troškovi održavanja, uz zadržavanje sličnog komunikacijskog dometa (oko 100 metara). Propusnost BLE-a je najčešće u rasponu od 150 kbps do 1 Mbps. [13]

Mobilni operacijski sustavi, uključujući iOS, Android, Windows Phone i BlackBerry, kao i macOS, Linux i Windows, imaju ugrađenu podršku za Bluetooth Low Energy.



Slika 9 BLE odašiljači (eng. *beacons*) [14]

#### 4.4. LoRa

LoRa (eng. *Long Range*, veliki domet) modulacijska je tehnologija širokopojasne mreže male snage. Koristi radiofrekvencije opsege ispod jednog gigaherca, koji se mogu koristiti bez licenciranja, poput EU433 (433,05 MHz – 434,79 MHz) i EU863-870 (863 MHz - 870/873 MHz) u Europi; AU915-928/AS923-1 (915 MHz - 928 MHz) u Australiji; US902-928 (902 MHz - 928 MHz) u Sjevernoj Americi; IN865-867 (865 MHz - 867 MHz) u Indiji; i AU915-928/AS923-1 i EU433 jugoistočnoj Aziji.

LoRa omogućuje prijenos na velike udaljenosti s niskom potrošnjom energije. LoRa tehnologija pokriva fizički sloj protokola, dok protokoli poput LoRaWAN -a (eng. *Long Range Wide Area Network* - mreža velikog raspona širokog dometa) pokrivaju više slojeve. Teoretska brzina prijenosa podataka iznosi između 0,3 kbps i 27 kbps, ovisno o faktoru širenja.

LoRaWAN definira komunikacijski protokol i arhitekturu sustava za mrežu, dok LoRa fizički sloj omogućuje komunikacijsku vezu velikog dometa. LoRaWAN je također odgovoran za kontroliranje napajanja, komunikacijskih frekvencija i brzina prijenosa podataka. Uređaji u mreži su asinkroni i emitiraju kada imaju na raspolaganju podatke za slanje. Podatke koje odašilje krajnji uređaj prima više pristupnika (eng. *gateway*), koji prosljeđuju podatkovne pakete na centralizirani mrežni poslužitelj. Podaci se zatim prosljeđuju aplikacijskim poslužiteljima. Tehnologija pokazuje visoku pouzdanost za umjereno opterećenje, međutim, ima nekih problema s performansama vezanim uz slanje potvrda.

LoRa uređaji imaju mogućnosti geolokacije, a teoretski domet iznosi više od 10 kilometara u idealnim uvjetima. <sup>[15]</sup>



Slika 10 Bosch LoRaWAN parkirni senzori <sup>[16]</sup>

## 5. Komponente sustava za kontrolu kvalitete zraka

Uređaj čija izrada je objašnjena u ovom radu očitavat će kvalitetu zraka u prostoriji, odnosno udio ugljičnog dioksida i razinu metalnih oksida, a ujedno će mjeriti temperaturu, tlak i vlažnost zraka. Za to će koristiti senzore CCS811 i BME280 i oni će predstavljati prvi sloj IoT arhitekture. Kao komunikacijska tehnologija odabran je Wi-Fi. Prikupljanje podataka sa senzora vršit će Wemos D1 Mini, razvojna pločica temeljena na modulu ESP8266, koji će preko Wi-Fi mreže slati podatke na MQTT broker koji je konfiguriran na Raspberry Pi računalu. Za dohvaćanje podataka sa brokera i prikaz rezultata mjerenja u Web aplikaciji koristit će se Python aplikacija, koja će koristiti Paho MQTT klijent, također konfiguriran na Raspberry Pi – ju. Navedene komponente opisane su u nastavku poglavlja.

### 5.1. CCS811 senzor kvalitete zraka

Kao glavni senzor sustava za kontrolu kvalitete zraka odabran je CCS811. To je digitalni senzor plinova vrlo niske potrošnje koji detektira širok raspon ukupnih isparljivih organskih spojeva (eng. *Total Volatile Organic Compounds* - TVOC), uključujući ekvivalent ugljičnog dioksida (eCO<sub>2</sub>) i razinu metalnih oksida (MOX). Isparljivi organski plinovi često su kategorizirani kao zagađivači zraka i/ili spojevi koji nadražuju osjetila, a mogu se pojaviti kao rezultat isparavanja iz raznih građevinskih materijala poput boja, kao i rezultat rada fotokopirnih uređaja te prisustva ljudi (disanje, pušenje). Ovaj senzor namijenjen je za uporabu u zatvorenim prostorima, te se koristi u pametnim uređajima poput smartphone-a i pametnih satova, ili u sustavima automatizacije domova. U ovom radu koristit će se izvedba senzora na zasebnoj tiskanoj pločici, koja za komunikaciju koristi standardnu I<sup>2</sup>C sabirnicu. Jedinствена I<sup>2</sup>C adresa je 0x5A<sub>HEX</sub>.



Slika 11 CCS811 senzor <sup>[17]</sup>



|                              |                    |
|------------------------------|--------------------|
| <b>Napon napajanja:</b>      | 1,8 V – 3,6 V      |
| <b>Jakost struje:</b>        | 30 mA              |
| <b>Potrošnja energije:</b>   | 60 mW (maksimalna) |
| <b>Napon logičke razine:</b> | 3,3 V – 5 V        |

Tablica 1 Karakteristike CCS811 senzora <sup>[18]</sup>

CCS811 sadrži sljedeće naponske pinove za priključivanje konektora napajanja:

- **VDD** – naponski pin za priključivanje na napon od 3,3 V.
- **GND** – zajedničko uzemljenje.

Objašnjenje logičkih pinova za spajanje konektora za komunikaciju:

- **SCL** – *clock* pin za I<sup>2</sup>C komunikaciju sa mikrokontrolerom. Preko njega se šalje *clock* (takt), odnosno vremenski signal koji sinkronizira komunikaciju senzora sa mikrokontrolerom. Na ovaj pin priključen je *pull-up* otpornik od 10 kΩ.
- **SDA** – podatkovni pin za I<sup>2</sup>C komunikaciju, spaja se sa I<sup>2</sup>C priključkom mikrokontrolera. Na ovaj pin također je priključen *pull-up* otpornik od 10 kΩ te se može koristiti logička razina u rasponu od 3 V – 5 V.
- **WAKE** – pin koji mora biti spojen sa uzemljenjem kako bi senzor bio u aktivnom stanju, ukoliko nije spojen sa uzemljenjem ulazi u stanje mirovanja.

I<sup>2</sup>C vrsta je komunikacijske sabirnice koja omogućava povezivanje *master* (upravljačkih) i *slave* (perifernih) uređaja. Kao što je već objašnjeno, koristi dvije linije za komunikaciju: SDA za slanje i primanje podataka i SCL za slanje *clocka*. Povezivanje više *slave* uređaja pomoću istih linija moguće je iz razloga što svaki uređaj ima jedinstvenu adresu.

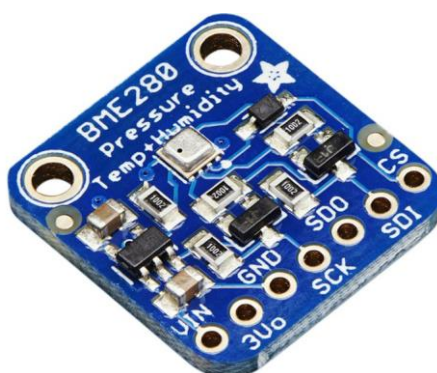
Svi periferni uređaj oslušuju SDA liniju i prate je li poslana njegova adresa. *Master* šalje prvi paket podataka koji sadrži startni bit i odgovarajuću adresu perifernog uređaja s kojim želi komunicirati. Periferni uređaj odgovora na poslani zahtjev za komunikaciju te nakon toga *master* počinje slati podatke. Kako bi se ostvarila komunikacija sa više *slave* uređaja koristi se *pull-up* otpornik. *Pull-up* je naziv za otpornik koji ima zadatak da „povuče“ obje linije, i SCL i SDA, na 5V, odnosno u logičku jedinicu. <sup>[19]</sup>

Za programiranje CCS811 senzora koristi se biblioteka *Adafruit\_CCS811*, kako bi se pojednostavio postupak programiranja i izbjeglo pisanje složenih algoritama. Napisani program naveden je u nastavku, zajedno s komentarima koji opisuju pojedine naredbe. Komentari se nalaze u redovima koji počinju sa znakom // i odnose se na naredbu ispod. <sup>[20]</sup>

```
#include "Adafruit_CCS811.h"
// stvaranje objekta imena ccs na temelju koristene biblioteke
Adafruit_CCS811 ccs;
void setup() {
  Serial.begin(9600);
  Serial.println("CCS811 test");
  // inicijalizacija senzora
  if(!ccs.begin(0x5A)){
    Serial.println("Pogreska kod pokretanja senzora CCS811!");
    while(1);
  }
  // Cekaj dok senzor ne bude spreman
  while(!ccs.available());
}
void loop() {
  if(ccs.available()){
    // slanje ocitanih vrijednosti
    if(!ccs.readData()){
      Serial.print("CO2: ");
      // koristenje geteCO2() metode
      Serial.print(ccs.geteCO2());
      Serial.print("ppm, TVOC: ");
      // koristenje getTVOC() metode
      Serial.println(ccs.getTVOC());
    }
    else{
      Serial.println("POGRESKA!");
      while(1);
    }
  }
  // ocitavanje se vrši svakih 500 ms
  delay(500);
}
```

## 5.2. BME280 senzor

Uz CCS811, koristit će se i BME280 senzor. To je kombinirani senzor za mjerenje temperature, relativne vlažnosti i barometarskog tlaka zraka. Veoma je malih dimenzija, niske potrošnje te se često koristi za kućnu automatizaciju, u prijenosnim GPS modulima ili u pametnim satovima. Ima visoku preciznost, pa je tako na temelju očitane vrijednosti tlaka moguće izračunati i približnu nadmorsku visinu. Kao i kod CCS811 senzora, koristit će se izvedba senzora na zasebnoj tiskanoj pločici. Za komunikaciju se također koristi standardna I<sup>2</sup>C sabirnica. Jedinствена I<sup>2</sup>C adresa je 0x76<sub>HEX</sub>.



Slika 12 BME280 senzor <sup>[21]</sup>

Karakteristike senzora navedene su u tablici:

|                                 |                                |
|---------------------------------|--------------------------------|
| <b>Napon napajanja:</b>         | 1,71 V – 3,6 V                 |
| <b>Jakost struje:</b>           | < 1 mA                         |
| <b>Napon logičke razine:</b>    | 3,3 V – 5 V                    |
| <b>Temperaturni raspon:</b>     | -40 °C – 85 °C (+/- 1 °C)      |
| <b>Raspon vlažnosti:</b>        | 0 – 100 % (+/- 3 %)            |
| <b>Raspon tlaka:</b>            | 300 hPa – 1100 hPa (+/- 1 hPa) |
| <b>Raspon nadmorske visine:</b> | 0 m – 9000 m (+/- 1 m)         |

Tablica 2 Karakteristike BME280 senzora <sup>[22]</sup>

Kao i CCS811 senzor, i BME280 sadrži dva naponska pina:

- **VDD** – naponski pin za priključivanje na napon od 3,3 V.

- **GND** – zajedničko uzemljenje.

Također, sadrži dva logička pina za spajanje konektora za komunikaciju:

- **SCL** – *clock* pin za sinkronizaciju I<sup>2</sup>C komunikacije sa mikrokontrolerom.
- **SDA** – podatkovni pin za I<sup>2</sup>C komunikaciju.

Za programiranje senzora koristi se biblioteka *Adafruit BME280*, kako bi se pojednostavio postupak programiranja. Napisani program naveden je u nastavku, zajedno s komentarima koji opisuju pojedine naredbe. Komentari se nalaze u redovima koji počinju sa znakom // i odnose se na naredbu ispod. <sup>[23]</sup>

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
// definiranje tlaka pri nultoj nadmorskoj visini
#define SEALEVELPRESSURE_HPA (1013.25)
// stvaranje objekta imena bme na temelju koristene biblioteke
Adafruit_BME280 bme;
void setup() {
    Serial.begin(9600);
    // inicijalizacija senzora
    if (!bme.begin(0x76)) {
        Serial.println("BME280 nije pronaden, provjerite spojeve!");
        while (1);
    }
}
void loop() {
    Serial.print("Temperatura = ");
    // koristenje readTemperature() metode
    Serial.print(bme.readTemperature());
    Serial.println("*C");

    Serial.print("Tlak = ");
    // koristenje readPressure() metode
    Serial.print(bme.readPressure() / 100.0F);
    Serial.println("hPa");

    Serial.print("Priblizna nadmorska visina = ");
    // koristenje readAltitude() metode
    Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
    Serial.println("m");

    Serial.print("Vlaznost = ");
```

```
// korištenje readHumidity() metode
Serial.print(bme.readHumidity());
Serial.println("%");

Serial.println();
// očitavanje se vrši svakih 1000 ms
delay(1000);
}
```

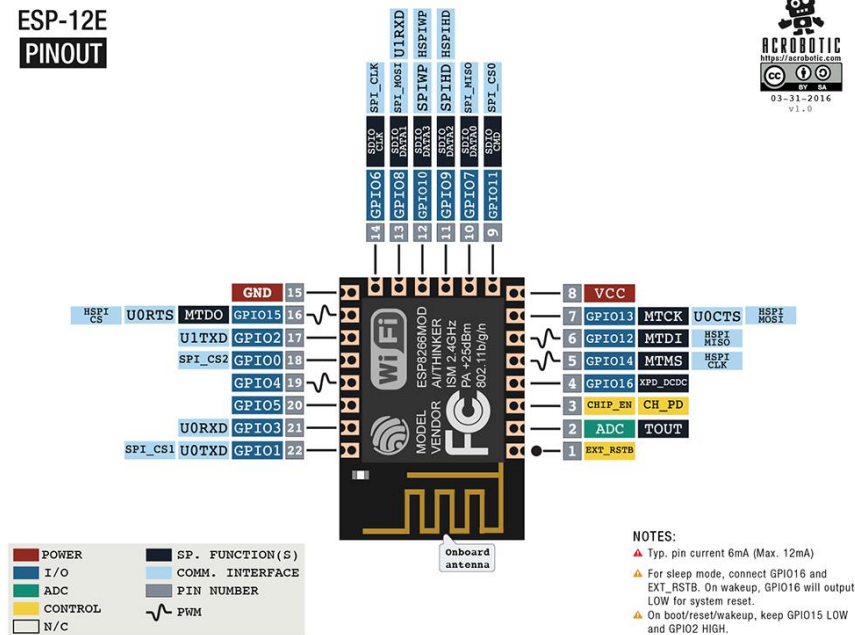
### 5.3. ESP8266

ESP8266 je 32-bitni Wi-Fi mikrokontroler vrlo niske cijene, proizvođača *Espressif Systems*, koji u potpunosti podržava TCP/IP protokol. Sadrži mikroprocesor koji radi na taktu od 80 MHz ili 160 MHz. Posjeduje memoriju od 32 KiB za upravljačke naredbe te 80 KiB za korisničke podatke. Radni napon i napon logičke razine je 3,3 V te ima 16 ulazno/izlaznih pinova. Može se koristiti kao mikrokontroler ili kao zasebni Wi-Fi modul kao nadopuna nekom drugom mikrokontroleru. Podržava IEEE 802.11 b/g/n Wi-Fi protokol. <sup>[24][25]</sup>



Slika 13 ESP8266 – 12E <sup>[26]</sup>

## ESP-12E PINOUT

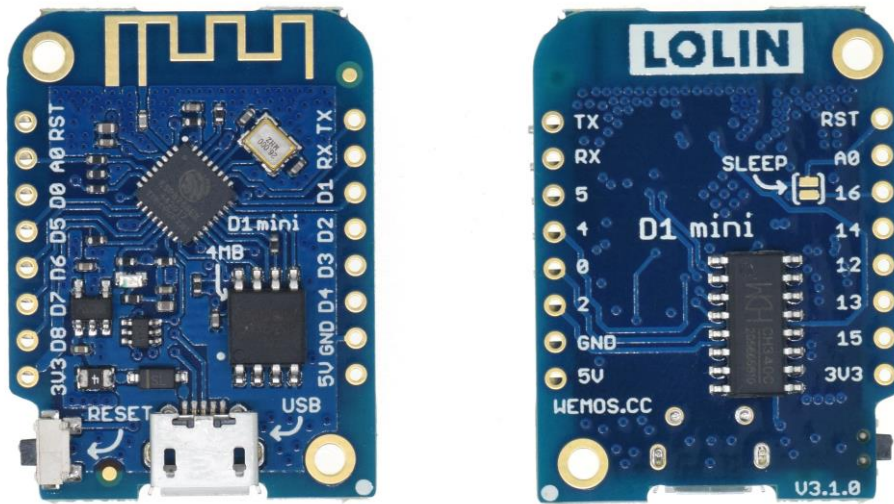


Slika 14 ESP-12E raspored pinova [27]

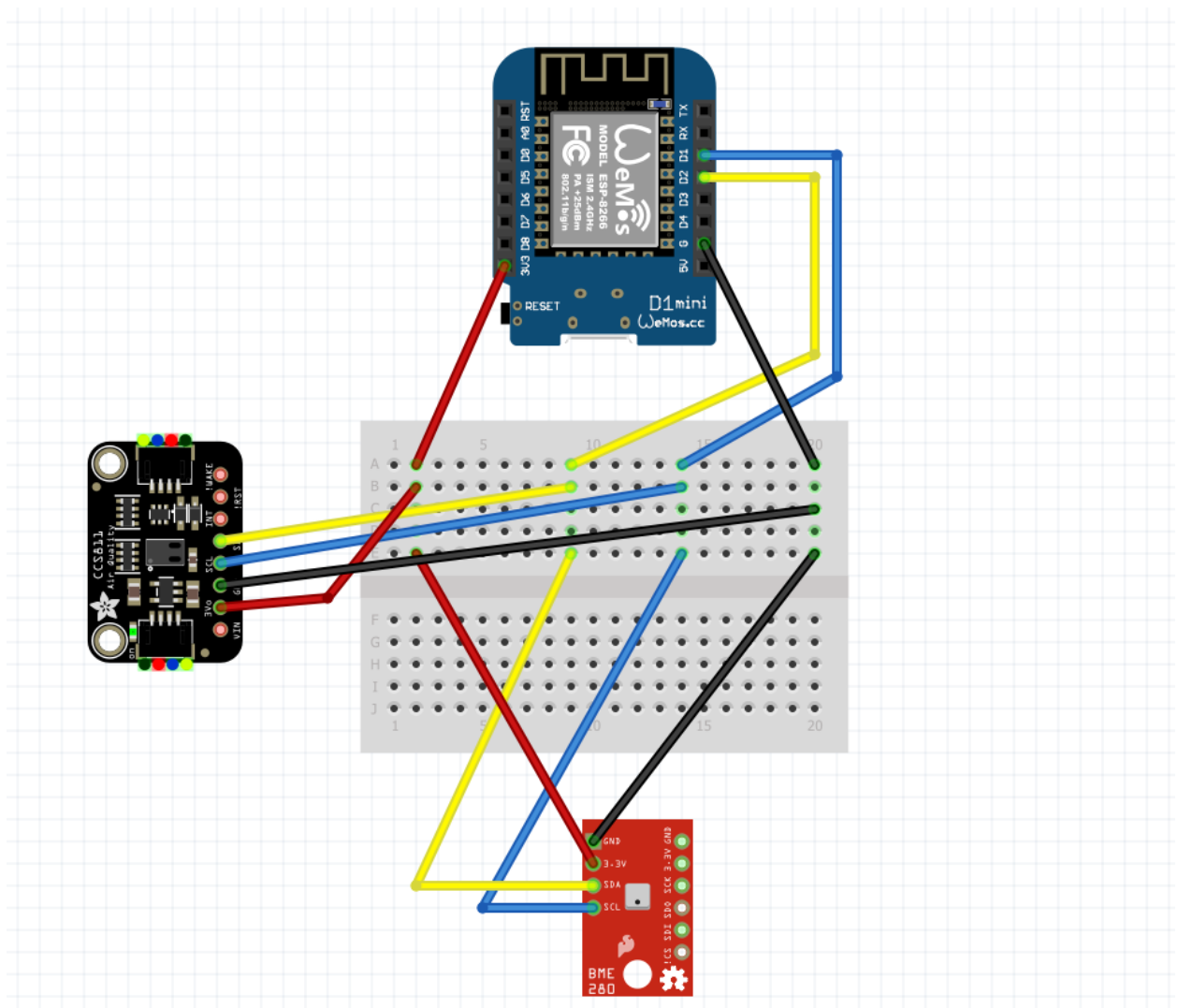
Budući da je ESP8266 mikrokontroler namijenjen za SMD lemljenje i radi na 3,3 V, kako bi se olakšalo spajanje sa računalom i pojednostavilo programiranje, u ovom radu koristi se razvojna pločica Wemos D1 Mini (u najnovijoj reviziji imena Lolin). Ova pločica bazirana je na ESP-12 verziji ESP8266 čipa i omogućuje jednostavno spajanje s računalom putem USB priključka te programiranje u Arduino IDE razvojnom okruženju. Zbog kombinacije mikrokontrolera i mogućnosti Wi-Fi povezivanja odličan je izbor za demonstraciju mogućnosti IoT platforme.

Procesor frekvencije 80 MHz na Wemos pločici ima na raspolaganju povećan kapacitet memorije za spremanje programskog koda, kapaciteta 1 MB, od čega je 0,8 MB dostupno, a 0,2 MB je rezervirano za *bootloader* te radnu memoriju kapaciteta 82 kB, od čega je korisniku dostupno 49 kB. Wemos D1 Mini ima podršku za I<sup>2</sup>C, SPI i serijsku komunikaciju, sadrži devet digitalnih ulazno/izlaznih pinova od kojih svi podržavaju PWM upravljanje te jedan analogni ulazni pin. Uz sve nabrojano, pločica koristi CH340 kao USB 2.0 sučelje, sadrži 3,3 voltni regulator i četiri LED indikatora. Prije spremanja programskog koda, Wemos vrši automatski hardverski reset. [28]

U ovom radu, Wemos pločica prikupljat će podatke sa senzora te ih putem Wi-Fi mreže bežično slati na Raspberry Pi, koristeći MQTT protokol.



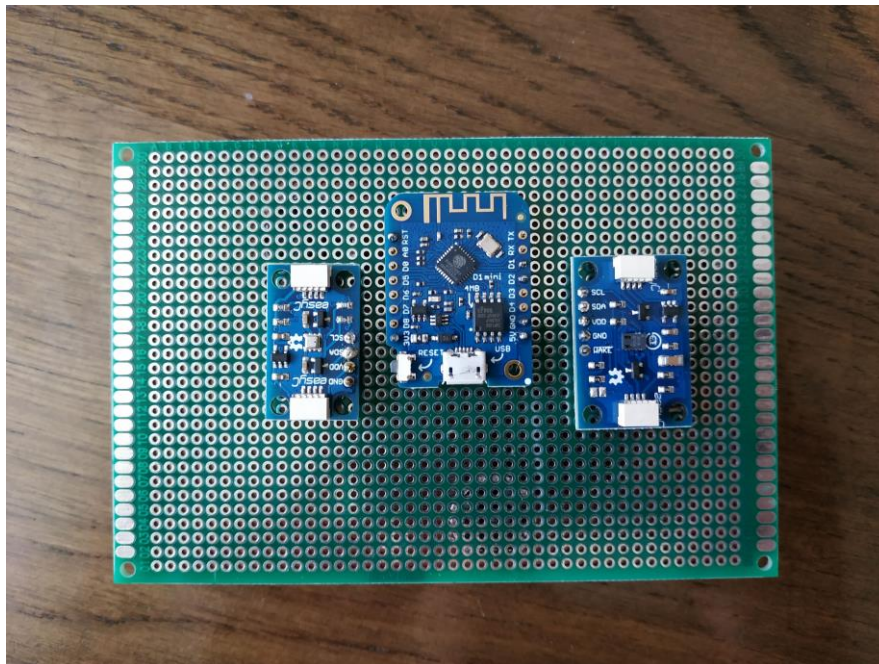
Slika 15 Wemos (Lolin) D1 Mini razvojna pločica [28]



Slika 16 Shema spajanja Wemosa D1 i senzora u programu „Fritzing“



Slika 16 prikazuje shemu spajanja senzora CCS811 i BME280 sa Wemosom D1 Mini pomoću eksperimentalne pločice.



Slika 17 Wemos D1 i senzori zalemljeni na bušenoj pločici

## 5.4. Raspberry Pi

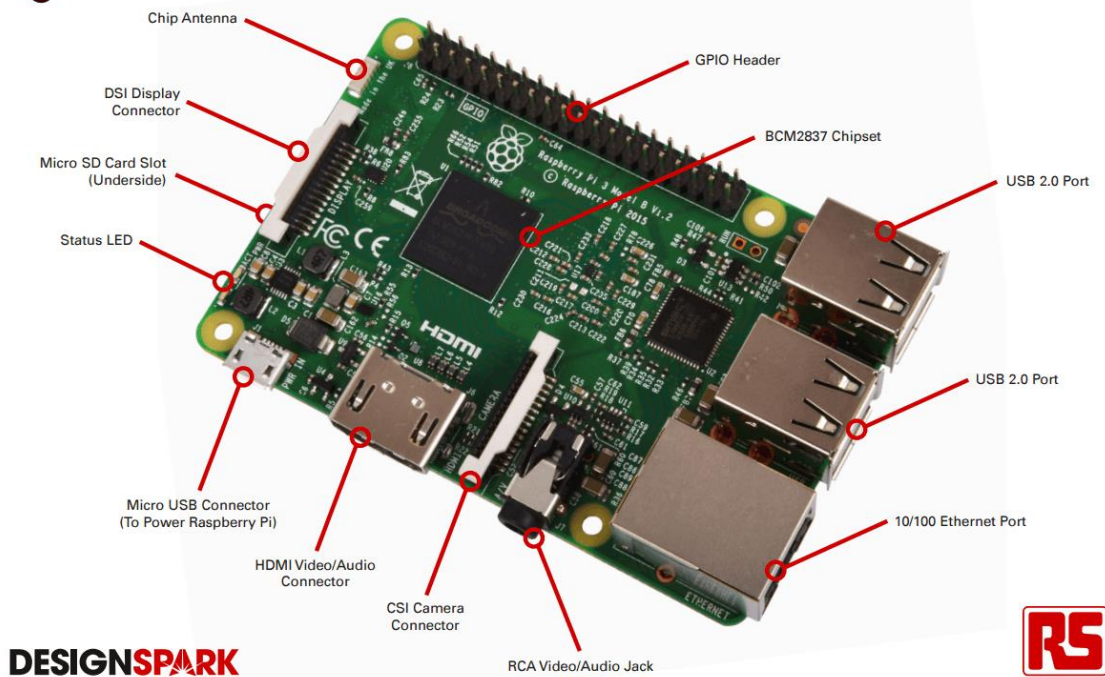
Raspberry Pi je mini računalo razvijeno od strane „*The Raspberry Pi Foundation*“, a osmišljen je kao jeftin i pristupačan alat za hobiste, entuzijaste, te za potrebe učenja programiranja u školama, pogotovo u slabije razvijenim zemljama. Vrlo je malih dimenzija, veličine kreditne kartice, te se sve komponente nalaze na jednoj tiskanoj pločici. Sadrži priključke i konektore za spajanje vanjskog napajanja, miša/tipkovnice te video izlaze za spajanje monitora. Operacijski sustav sprema se na microSD karticu, a u ovom radu koristi se Raspbian operacijski sustav, baziran na Linuxu.

Slika 18 pokazuje Raspberry Pi 3B, te funkcije njegovih priključaka.





## Raspberry Pi 3 Model B



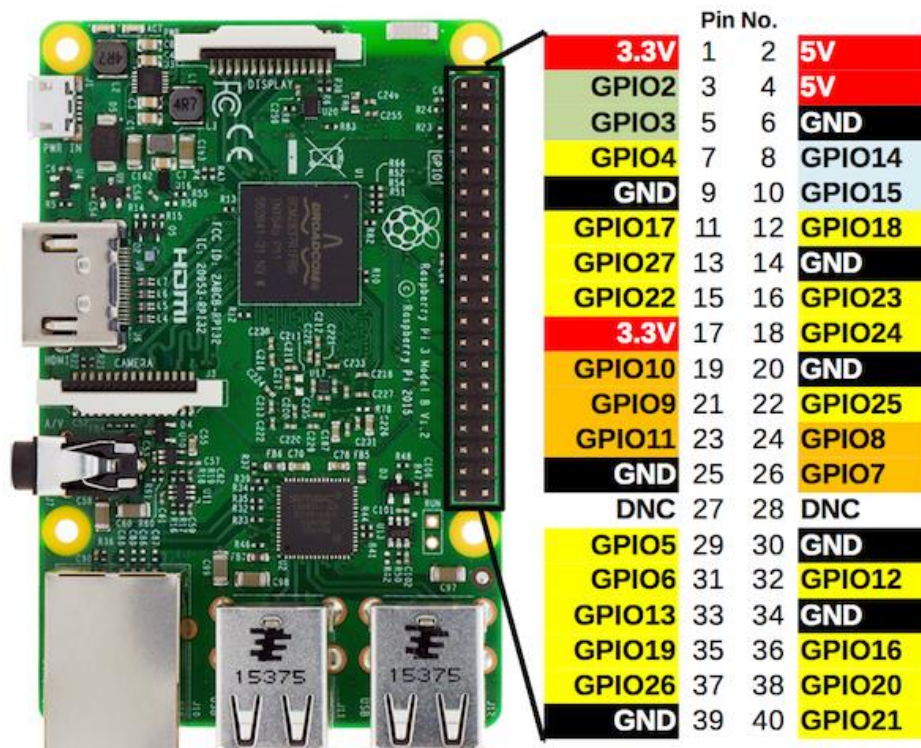
DESIGNSPARK



Slika 18 Konektori na Raspberry Pi 3B modelu <sup>[29]</sup>

Raspberry Pi 3B, kakav se koristi u ovom radu, sadrži četrdeset GPIO (eng. *General Input Output Pin*, standardni ulazno – izlazni pinovi) pinova. GPIO pinovi dijele se dijele na pinove s konstantnim naponom od 5,0 V, s konstantnim naponom od 3,3 V, pinove za uzemljenje (eng. *Ground - GND*), na pinove sa podesivim naponom te na pinove na koje se ne spaja ništa (eng. *Do Not Connect – DNC*). <sup>[30]</sup>

Slika 19 prikazuje raspored GPIO pinova na tiskanoj pločici Raspberry Pi računala.



Slika 19 Raspored GPIO pinova na Rasperry Pi pločici [31]

Na Raspberry Pi mini računalu bit će konfiguriran Mosquitto MQTT broker i klijent koji će omogućavati lokalni pristup Web aplikaciji za prikaz očitavanja vrijednosti sa senzora. Poglavlje 6 detaljnije opisuje MQTT protokol.

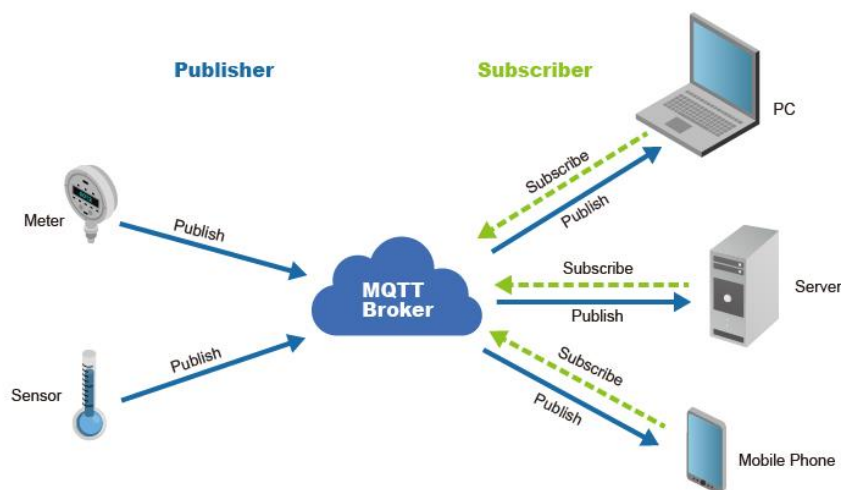


Slika 20 Rasperry Pi 3B u kućištu

## 6. MQTT protokol

MQTT (engl. *Message Queuing Telemetry Transport*) komunikacijski je protokol koji se često koristi u IoT projektima. Zadužen je za upravljanje prijenosom podataka i bazira se na objavi/pretplati (engl. *publish/subscribe*) principu rada. Koristi se kod prijenosa male količine podataka na udaljenim lokacijama, ili kod slučajeva gdje je dostupna mala propusnost bežičnog prijenosa podataka. Jednostavan je i dizajniran je da se jednostavno implementira. Obično radi preko postojećeg TCP/IP protokola, ali kao osnovu može koristiti i druge dvosmjerne protokole sa zaštitom protiv gubitaka (npr. UDP protokol). [32]

Arlen Nipper i Andy Stanford-Clark kreirali su prvu verziju MQTT protokola 1999. godine. IBM je izdao sljedeću verziju (v3.1) 2013. godine, a zatim i verziju v3.1.1 koja je postala OASIS (eng. *Organization for the Advancement of Structured Information Standards*) standard. MQTT protokol počeo je koristiti ISO standard 2016. godine (ISO/IEC PRF 20922).



Slika 21 Princip rada MQTT protokola [33]

Slika 21 prikazuje osnovni princip rada MQTT protokola. MQTT broker komunicira sa svim klijentima (uređajima). S lijeve strane su klijenti koji mjere određene veličine i objavljuju te podatke, a s desne strane su klijenti koji vrše obradu podataka na koje su pretplaćeni. MQTT protokol radi na principu objavi/pretplati (eng. *publish/subscribe*) metode. Senzori koji mjere neke fizikalne veličine objavljuju (eng. *publish*) rezultate mjerenja na određenu temu (eng. *topic*). Klijenti kojima je potrebno dostaviti informacije su pretplaćeni (eng. *subscribe*) na tu istu temu. Kada je objavljena poruka na određenoj temi, nju dobivaju svi klijenti koji su se pretplatili na tu temu.

## 6.1. MQTT broker

MQTT broker obavlja zadaću „posrednika“ između klijenata te je time i središnji dio cijelog objavi/pretplati mehanizma. Glavne zadaće MQTT brokera su: primanje poruka od svih klijenata koji objavljuju na određenu temu, filtriranje poruka, odlučivanje o prosljeđivanju određenih poruka te slanje poruka pretplaćenim klijentima.

Posao MQTT brokera može obavljati svako računalo koje je specifikacijama dovoljno moćno za odabrani broker, bilo da je temeljeno na Windows ili Linux operacijskom sustavu. Danas postoji velik broj takvih programa. Neki od najčešće korištenih besplatnih MQTT brokera su Mosquitto, Emqtt, Hive MQ, MOSCA.

U ovom diplomskom radu, kao broker koristi se Mosquitto, koji je konfiguriran na Raspberry Pi 3B računalu.

## 6.2. MQTT klijent

MQTT klijenti obuhvaćaju pretplatnike i objavitelje. MQTT klijent može istovremeno imati funkciju pretplatnika (klijent pretplaćen na temu) i objavitelja (klijent objavljuje poruku na temu). Zadaću MQTT klijenta može obavljati širok raspon uređaja, od mikrokontrolera do osobnih računala, odnosno svaki uređaj koji ima implementiranu podršku za rad s MQTT protokolom (MQTT knjižnice). MQTT programske knjižnice (eng. *libraries*) dostupne su za velik broj programskih jezika: Java, JavaScript, C, C++, C#, kao i za platforme iOS i .NET.

## 6.3. Razina kvalitete usluge kod MQTT protokola

Kvaliteta usluge (engl. *Quality of Service, QoS*) je dogovor (sporazum) između objavitelja i pretplatnika na poruku koji definira sigurnost isporuke poruke, ovisno o razini dogovora.

Kod MQTT protokola, definirane su tri razine sporazuma:

1. razina 0: najviše jednom (engl. *At most once*) – poruka je poslana samo jedanput, ni klijent ni broker ne vrše nikakve dodatne provjere je li poruka uspješno primljena,
2. razina 1: barem jedanput (engl. *At least once*) – poruka se šalje tako dugo dok klijent ne potvrdi njezin primitak,
3. razina 2: točno jedanput (engl. *Exactly once*) – broker i klijent obavljaju razmjenu poruka u dvije razine (eng. *two level handshake*), kako bi bili sigurni da je poruka stigla do klijenta točno jedanput.

## 7. Programiranje Raspberry Pi mini računala

### 7.1. Konfiguriranje MQTT brokera

Za korištenje MQTT protokola, potrebno je konfigurirati MQTT broker, kao što je već objašnjeno u prethodnim poglavljima. Za potrebe ovog rada, koristit će se Mosquitto broker na Raspberry Pi mini računalu. U ovom poglavlju bit će objašnjeno njegovo konfiguriranje. Prije instalacije Mosquitto, obaviti će se softverska nadogradnja Raspberry Pi-ja. <sup>[34][35]</sup>

```
sudo apt-get update
sudo apt-get upgrade
```

Nakon toga, instalira se Mosquitto i njegov pripadni klijentski paket.

```
sudo apt-get install mosquitto -y
sudo apt-get install mosquitto-clients -y
```

Nakon instalacije ova dva paketa, potrebno je konfigurirati broker. Konfiguracijska datoteka Mosquitto brokera nalazi se na putanji at /etc/mosquitto/mosquitto.conf.

```
sudo nano /etc/mosquitto/mosquitto.conf
```

Na dnu datoteke nalazi se linija:

```
include_dir /etc/mosquitto/conf.d
```

Ona će se izbrisati te će se dodati ove tri linije, koje govore Mosquitto da se jedino klijenti koji imaju ispravno korisničko ime i lozinku mogu pretplatiti na njegove teme, te da Mosquitto sluša poruke na portu 1883:

```
allow_anonymous false
password_file /etc/mosquitto/pwfile
listener 1883
```

Ako korisnik ne želi da broker zahtijeva korisničko ime i lozinku od pretplatničkih klijenata, izostavlja prve dvije linije. Spremaju se promjene u datoteci i ona se zatvara. Nakon toga, potrebno

je odrediti korisničko ime i lozinku. Upiše se sljedeća naredba: - replacing username, gdje je username željeno korisničko ime. Zatim se upiše željena lozinka te se obavi ponovno pokretanje Raspberry Pi-ja naredbom:

```
sudo reboot
```

Mosquitto MQTT broker sada je konfiguriran te je spreman za primanje poruka sa Wemos D1 mikrokontrolerske pločice.

## 7.2. Python aplikacija za prikaz rezultata

Za prikaz vrijednosti sa senzora, koje Raspberry Pi dobije sa Wemosom putem MQTT protokola, napravit će se Python aplikacija koja koristi Paho MQTT klijent, te će dobivene rezultate prikazati u Web aplikaciji, kojoj se može pristupiti unutar lokalne mreže. Web aplikacija bit će jednostavnog dizajna, prilagođena za prikaz na pametnim mobitelima. Funkcija koja je zadužena za generiranje sadržaja Web aplikacije navedena je u nastavku.

```
String SendHTML(float co2, float tvoc, float temp, float tlak, float
vlaznost, float visina)
{
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width,
initial-scale=1.0, user-scalable=no\">\n";
    ptr += "<title>Kontrola kvalitete zraka - UNIN</title>\n";
    ptr += "<style>html { font-family: Helvetica; display: inline-block;
margin: 0px auto; text-align: center;}\n";
    ptr += "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto
30px;}\n";
    ptr += "p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";
    ptr += "</style>\n";
    ptr += "<script>\n";
    ptr += "setInterval(loadDoc,1000);\n";
    ptr += "function loadDoc() {\n";
    ptr += "document.body.innerHTML =this.responseText}\n";
    ptr += "};\n";
    ptr += "</script>\n";
    ptr += "</head>\n";
    ptr += "<body>\n";
    ptr += "<div id=\"webpage\">\n";
```

```
ptr += "<h1>Kontrola kvalitete zraka - UNIN</h1>\n";

ptr += "<p>CO2: ";
ptr += co2;
ptr += " ppm</p>";

ptr += "<p>TVOC: ";
ptr += tvoc;
ptr += " ppb</p>";

ptr += "<p>Temperatura: ";
ptr += temp;
ptr += " °C</p>";

ptr += "<p>Tlak: ";
ptr += tlak ;
ptr += " hPa</p>";

ptr += "<p>Relativna vlaznost: ";
ptr += vlaznost;
ptr += " %</p>";

ptr += "<p>Nadmorska visina: ";
ptr += visina;
ptr += " m</p>";

ptr += "</div>\n";
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}
```

## 8. Programiranje Wemos D1 Mini

U poglavlju 5 navedeni su programski kodovi za senzore CCS811 i BME280 koji šalju rezultate mjerenja na serijski ulaz Wemos D1 Mini pločice. U ovom poglavlju bit će objašnjen kod za Wemos koji vrijednosti sa oba senzora šalje preko lokalne Wi-Fi mreže na Raspberry Pi, koji je konfiguriran u prošlom poglavlju, koristeći MQTT protokol.

```
#include <Wire.h>
#include "Adafruit_CCS811.h"
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include "PubSubClient.h" // biblioteka za spajanje i slanje na MQTT broker

// Kod za ESP8266
// #include "ESP8266WiFi.h" // omogućava Wemosu spajanje na lokalnu mrežu (putem
// Wi-Fi-ja)
// #define DHTPIN D1

// WiFi
const char* ssid = "unin-dipl-guest"; // Mrežni SSID
const char* wifi_password = "123456"; // Wi-Fi lozinka

// MQTT
const char* mqtt_server = "192.168.42.10"; // IP adresa MQTT brokera (Raspberry
// Pi)
const char* co2_topic = "dipl/co2";
const char* tvoc_topic = "dipl/tvoc";
const char* temp_topic = "dipl/temp";
const char* vlaznost_topic = "dipl/vlaznost";
const char* tlak_topic = "dipl/tlak";
const char* nadm_visina_topic = "dipl/nadm_visina ";

const char* mqtt_username = "gdelija"; // MQTT username
const char* mqtt_password = "1234"; // MQTT password
const char* clientID = "sipl_sensors"; // MQTT client ID

// Inicijalizacija objekata za WiFi i MQTT klijenta
WiFiClient wifiClient;
// 1883 je port za slusanje Broker
PubSubClient client(mqtt_server, 1883, wifiClient);
```



Na početku programskog koda, unosimo biblioteke za korištenje senzora CCS811 i BME280, kao i „PubSubClient.h“ biblioteku za korištenje MQTT protokola na Wemos D1 pločici. „ESP8266WiFi.h“ biblioteka omogućuje Wemosu spajanje na Wi-Fi mrežu. Definiiraju se teme (eng. *topics*) koje će se slati putem MQTT-a, kao i korisničko ime i lozinka koji moraju odgovarati onima koji su podešeni u Mosquitto brokeru na Raspberry Pi-ju.

```
// Funkcija za spajanje Wemos na MQTT broker
void connect_MQTT(){
    Serial.print("Connecting to ");
    Serial.println(ssid);

    // Spoji se na Wi-Fi
    WiFi.begin(ssid, wifi_password);

    // Čekanje dok se ne uspostavi Wi-Fi veza prije nastavka programa
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // Debugging - ispis IP adrese Wemos D1
    Serial.println("WiFi spojen");
    Serial.print("IP adresa: ");
    Serial.println(WiFi.localIP());

    // Spajanje na MQTT broker
    if (client.connect(clientID, mqtt_username, mqtt_password)) {
        Serial.println("Uspješno spajanje na MQTT broker!");
    }
    else {
        Serial.println("Neuspješno spajanje na MQTT broker...");
    }
}
```

Funkcija za spajanje Wemos D1 na MQTT broker i slanje podataka. Koriste se ranije definirani korisničko ime i lozinka. U slučaju nemogućnosti povezivanja, korisniku vraća upozorenje.

```
// Inicijalizacija senzora
void setup() {
```

```

Serial.begin(9600);
if(!ccs.begin(0x5A)){
    Serial.println("Pogreska kod pokretanja senzora CCS811!");
    while(1);
}
if (!bme.begin(0x76)) {
    Serial.println("Pogreska kod pokretanja senzora BME280!");
    while (1);
}}

```

Ovaj dio koda inicijalizira senzore CCS811 i BME280.

```

void loop() {
    connect_MQTT();
    Serial.setTimeout(2000);

    if(!ccs.readData()){
        Serial.print("CO2: ");
        float co2 = ccs.getCO2();
        Serial.print("ppm, TVOC: ");
        float tvoc = ccs.getTVOC();
    }

    float temp = bme.readTemperature();
    float tlak = bme.readPressure() / 100.0F;
    float vlaznost = bme.readHumidity();
    float nadm_visina = bme.readAltitude(SEALEVELPRESSURE_HPA);

    Serial.print("CO2: ");
    Serial.print(co2);
    Serial.println(" ppm");
    Serial.print("TVOC: ");
    Serial.print(tvoc);
    Serial.println(" ppb");
    Serial.print("Temp: ");
    Serial.print(temp);
    Serial.println(" *C");
    Serial.print("Tlak: ");
    Serial.print(tlak);
    Serial.println(" hPa");
    Serial.print("Vlaznost: ");
    Serial.print(vlaznost);
}

```

```
Serial.println(" %");
Serial.print("Nadm. visina: ");
Serial.print(nadm_visina);
Serial.println(" m");
```

Korištenje ugrađenih funkcija iz biblioteka senzora, pomoću kojih se dobivaju izmjerene vrijednosti sa senzora i njihovo spremanje u varijable kako bi se u daljnjem dijelu koda mogle slati pomoću MQTT-a.

```
// MQTT moze slati samo stringove
String co2s="CO2: "+String((float)co2)+" ppm ";
String tvocs="TVOC: "+String((float)tvoc)+" ppb ";
String temps="Temp: "+String((float)temp)+" *C ";
String tlaks="Tlak: "+String((float)tlak)+" hPa ";
String vlaznosts="Vlaznost: "+String((float)vlaznost)+" % ";
String nadm_visinas="Nadm. visina: "+String((float)nadm_visina)+" m ";

// PUBLISH na MQTT Broker (topic = CO2, definirana na početku)
if (client.publish(co2_topic, String(co2s).c_str())) {
    Serial.println("CO2 vrijedn. poslana!");
}

// Pokušaj ponovno ako poruka nije uspješno poslana.
else {
    Serial.println("CO2 nije uspješno poslan. Ponovno spajanje na MQTT
Broker.");
    client.connect(clientID, mqtt_username, mqtt_password);
    delay(10);
    client.publish(co2_topic, String(co2s).c_str());
}

// PUBLISH na MQTT Broker (topic = TVOC, definirana na početku)
if (client.publish(tvoc_topic, String(tvocs).c_str())) {
    Serial.println("TVOC vrijedn. poslana!");
}

// Pokušaj ponovno ako poruka nije uspješno poslana.
else {
    Serial.println("TVOC nije uspješno poslan. Ponovno spajanje na MQTT
Broker.");
    client.connect(clientID, mqtt_username, mqtt_password);
```

```

    delay(10);
    client.publish(tvoc_topic, String(tvocs).c_str());
}

// PUBLISH na MQTT Broker (topic = temperatura, definirana na početku)
if (client.publish(temp_topic, String(temps).c_str())) {
    Serial.println("Temperaturna vrijedn. poslana!");
}

// Pokušaj ponovno ako poruka nije uspješno poslana.
else {
    Serial.println("Temperatura nije uspješno poslana. Ponovno spajanje na
MQTT Broker.");
    client.connect(clientID, mqtt_username, mqtt_password);
    delay(10);
    client.publish(temp_topic, String(temps).c_str());
}

// PUBLISH na MQTT Broker (topic = tlak, definirana na početku)
if (client.publish(tlak_topic, String(tlaks).c_str())) {
    Serial.println("Vrijedn. tlaka poslana!");
}

// Pokušaj ponovno ako poruka nije uspješno poslana.
else {
    Serial.println("Tlak nije uspješno poslan. Ponovno spajanje na MQTT
Broker.");
    client.connect(clientID, mqtt_username, mqtt_password);
    delay(10);
    client.publish(tlak_topic, String(tlaks).c_str());
}

// PUBLISH na MQTT Broker (topic = vlaznost, definirana na početku)
if (client.publish(vlaznost_topic, String(vlaznosts).c_str())) {
    Serial.println("Vlaznost poslana!");
}

// Pokušaj ponovno ako poruka nije uspješno poslana.
else {
    Serial.println("Vlaznost nije uspješno poslana. Ponovno spajanje na MQTT
Broker.");
    client.connect(clientID, mqtt_username, mqtt_password);
}

```

```

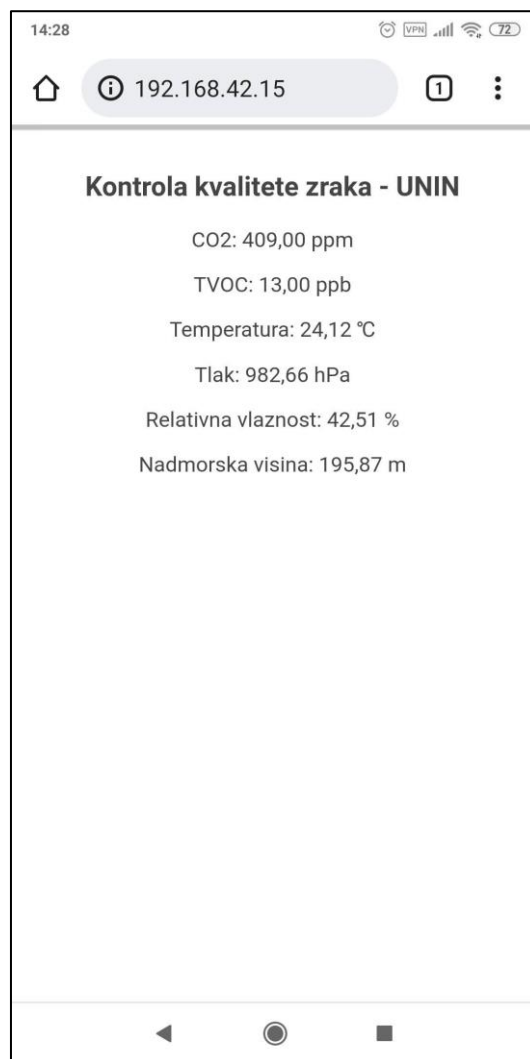
    delay(10);
    client.publish(vlaznost_topic, String(vlaznosts).c_str());
}

// PUBLISH na MQTT Broker (topic = nadm_visina, definirana na početku)
if (client.publish(nadm_visina_topic, String(nadm_visinas).c_str())) {
    Serial.println("Nadm. visina poslana!");
}
// Pokušaj ponovno ako poruka nije uspješno poslana.
else {
    Serial.println("Nadm. visina nije uspješno poslana. Ponovno spajanje na
MQTT Broker.");
    client.connect(clientID, mqtt_username, mqtt_password);
    delay(10);
    client.publish(nadm_visina_topic, String(nadm_visinas).c_str());
}

client.disconnect();
delay(1000*60);
}

```

MQTT može slati podatke samo u *string* obliku (znakovnom). Stoga je prvo potrebno konvertirati izmjerene vrijednosti sa senzora iz realnog (*float*) oblika u znakovni zapis. Zatim se radi slanje (*Publish*) vrijednosti na MQTT broker. Na kraju koda obavlja se otpajanje sa MQTT topica.



Slika 22 Web aplikacija otvorena na pametnom mobitelu

## 9. Mjerenje kvalitete zraka u realnim uvjetima

Korištenjem izrađenog uređaja, napravljena su mjerenja zraka u tri različite okoline:

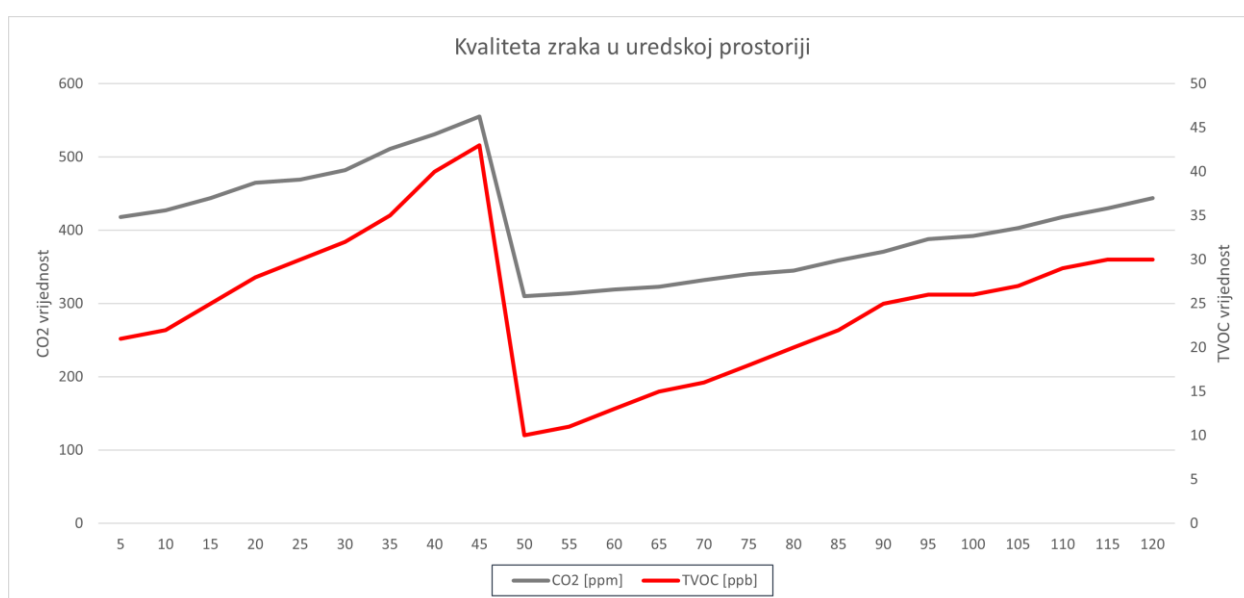
1. uredska prostorija,
2. podzemna garaža,
3. industrijska hala za zavarivanje.

Senzor je očitavao vrijednosti svakih pet minuta u trajanju od dva sata, a prije početka mjerenja radio je jedan sat kako bi mjerenje bilo što točnije (u dokumentaciji CCS811 senzora preporuča se barem pola sata „zagrijavanja“ (eng. *warm-up time*)).

Vrijednosti veličina koje mjeri CCS811 senzor i njihov utjecaj na ljude prikazan je u tablici 3.

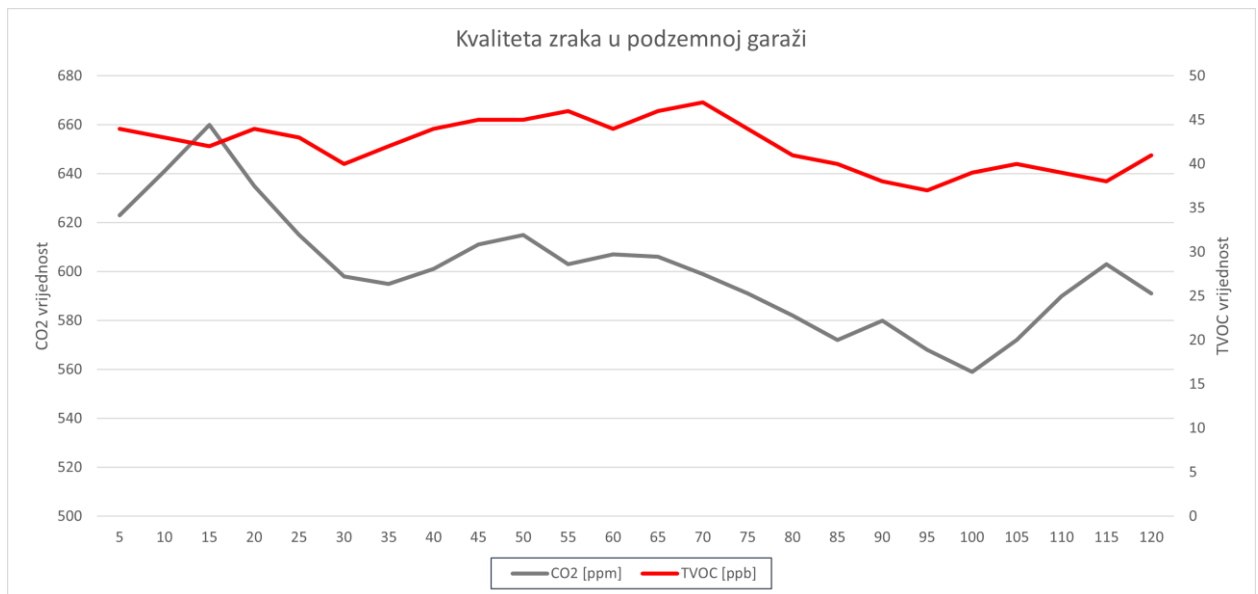
| Koncentracija CO <sub>2</sub><br>[ppm] | Utjecaj na ljude   | Koncentracija<br>TVOC-a [ppb] | Utjecaj na ljude                     |
|--|--------------------|-------------------------------|--------------------------------------|
| < 500                                  | normalno           | < 50                          | normalno                             |
| 500 - 1000                             | malo neugodno      | 50 - 750                      | neugodno, tjeskoba                   |
| 1000 - 2500                            | umor               | 750 - 6000                    | glavobolja, depresija                |
| 2500 - 5000                            | štetno za zdravlje | > 6000                        | glavobolja, štetno za živčani sustav |

Tablica 3 Vrijednosti koncentracije CO<sub>2</sub> i TVOC-a i utjecaj na ljude <sup>[18]</sup>



Grafikon 1 Izmjerena kvaliteta zraka u uredskoj prostoriji

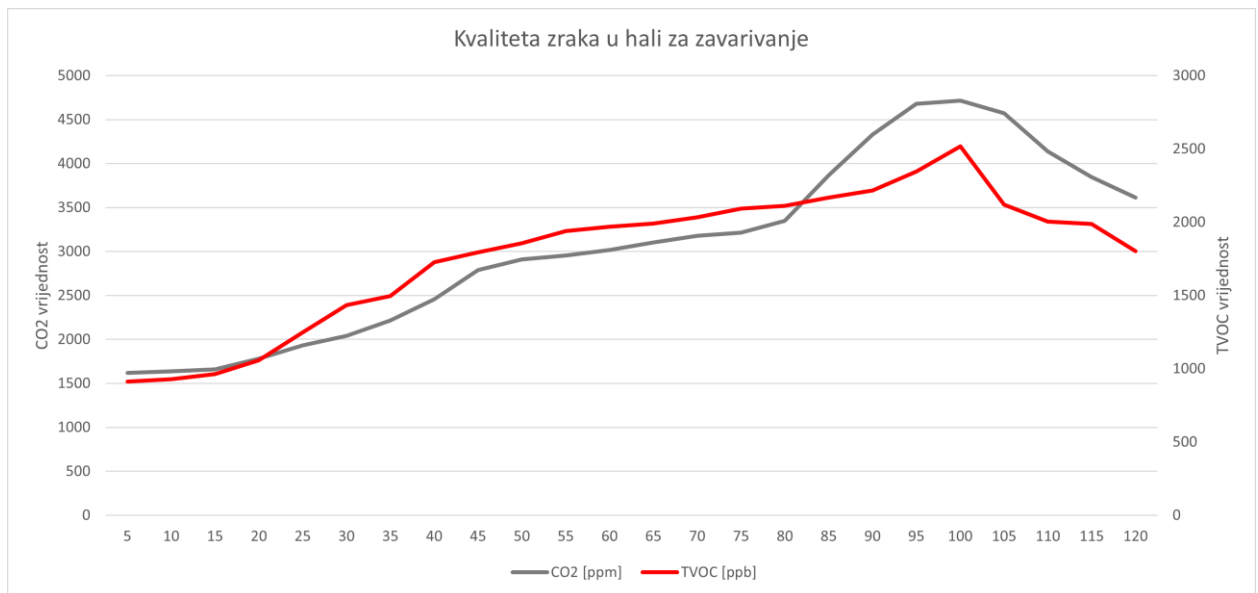
Prvo mjerenje napravljeno je u uredskoj prostoriji sa dvadesetak zaposlenika. Vrijednost CO<sub>2</sub> kretala se u okvirnom rasponu od 300 ppm-a do 550 ppm-a, dok se vrijednost TVOC-a kretala u rasponu od 10 ppb-a do 45 ppb-a. Nakon 45 minuta od početka mjerenja, otvoren je prozor te je prostorija provjetrena, što je doprinijelo smanjenju koncentracije obje veličine. Izmjerene koncentracije su u granicama normale te ne izazivaju ikakve smetnje i neugodu kod ljudi.



Grafikon 2 Izmjerena kvaliteta zraka u podzemnoj garaži

Drugo mjerenje obavljeno je na drugoj razini podzemne garaže. Vrijednost CO<sub>2</sub> kretala se u rasponu od 560 ppm-a do 660 ppm-a, dok se koncentracija TVOC-a cijelo vrijeme kretala u rasponu od 40 ppb-a do 45 ppb-a. Vrijednost CO<sub>2</sub> je malo iznad normalne, ali još uvijek nije izazivala nikakve smetnje, dok je vrijednost TVOC-a u granicama normale. Koncentracije su cijelo vrijeme bile na sličnoj razini zbog stalne prisilne izmjene zraka u podzemnoj garaži.





Grafikon 3 Izmjerena kvaliteta zraka u hali za zavarivanje

Treće mjerenje obavljeno je u industrijskoj hali u kojoj rade petorica zavarivača te još dvojica zaposlenika na ostalim poslovima (bušenje, rezanje, brušenje metala). Koncentracija CO<sub>2</sub> kretala se u rasponu od 1500 ppm-a do 4750 ppm-a, dok su vrijednosti TVOC-a bile između 900 ppb-a i 2500 ppb-a. Izmjerene koncentracije su vrlo visoke zbog poslova koji u zrak donose veliku količinu metalnih čestica, kao i maziva, boja i otapala. Ovakva radna okolina zahtijeva upotrebu zaštitnih maska za disanje (respiratora), jer je ovoliko povišena koncentracija tvari u zraku vrlo štetna za zdravlje zaposlenika.

Obavljanjem mjerenja, dobivena je ideja za moguće poboljšanje uređaja, a to je automatsko slanje e-mail poruke ili SMS-a odgovornoj osobi u tvrtki kada koncentracije CO<sub>2</sub> i TVOC-a premaše određenu vrijednost, kako bi se provjetrile prostorije.

## 10. Zaključak

Na primjeru senzora kvalitete zraka objašnjen je koncept Interneta stvari. Budući da je IoT relativno novi pojam i platforma, za pretpostaviti je da će u bližoj budućnosti većina proizvoda na tržištu postati dijelom te platforme, te da će imati mogućnost praćenja ili upravljanjem nekih od svojih parametara. IoT kao industrija ima ogroman potencijal za razvitak.

Obrađeni uređaj za kontrolu kvalitete zraka ima široku potencijalnu primjenu, od industrijskih postrojenja do kućnih sustava automatizacije. Relativno jeftini i lako dostupni dijelovi omogućuju kontrolu i obradu raznih veličina, te njihovo spremanje u bazu podataka za mogućnost provjere vrijednosti u nekom prošlom vremenskom periodu i usporedbu sa današnjim. Tako bi se npr. mogla napraviti usporedba je li neka promjena u proizvodnom procesu neke tvrtke doprinijela poboljšanju kvalitete zraka u zatvorenim prostorima, što ima za rezultat ugodnije radno mjesto zaposlenicima i smanjenje štetnog utjecaja na njihovo zdravlje.

Razvoj pametnih uređaja olakšan je upotrebom razvojnih pločica baziranih na raznim mikrokontrolerima, poput korištenog Wemosa D1 Mini. Takve razvojne pločice na sebi sadrže već mnoge konektore i pinove koji olakšavaju eksperimentiranje i testiranje sa raznim sensorima i drugom elektronikom. Naravno, uređaji koji se rade u masovnoj proizvodnji koriste samo mikrokontrolere i one elektroničke komponente koje su potrebne, ali za entuzijaste i one koji se žele upoznati sa elektronikom i programiranjem, razvojne pločice su odlično rješenje.

Pokazano je kako jeftina mini računala poput Raspberry Pi-ja imaju dovoljno snage za pokretanje servera koji vrše kontrolu nad manjim IoT sustava. Takva računala svake godine dolaze sa sve boljim specifikacijama i manjom cijenom i teško je predvidjeti do kamo će dogurati za nekoliko godina i što će sve biti sposobna pokretati.



IZJAVA O AUTORSTVU  
I  
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, GORAN DELIJA (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/~~ica~~ završnog/diplomskog (obrisati nepotrebno) rada pod naslovom KONTROLA KVALITETE ZRAKA KORIŠTENJEM (upisati naslov) te da u navedenom radu ~~nisu~~ nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:  
(upisati ime i prezime)

Goran Delija  
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, GORAN DELIJA (ime i prezime) neopozivo izjavljujem da sam suglasan/~~na~~ s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom KONTROLA KVALITETE ZRAKA (upisati naslov) čiji sam autor/~~ica~~ KORIŠTENJEM IOT PLATFORME

Student/ica:  
(upisati ime i prezime)

Goran Delija  
(vlastoručni potpis)

## 11. Literatura

- [1] <https://medium.com/swlh/internet-of-things-iot-d4fce78a1b70> (24.08.2021.)
- [2] [https://miro.medium.com/max/860/1\\*vUFNgofROTwY\\_rGd4\\_7Qsw.jpeg](https://miro.medium.com/max/860/1*vUFNgofROTwY_rGd4_7Qsw.jpeg) (24.08.2021.)
- [3] [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things) (24.08.2021.)
- [4] <https://iot-analytics.com/internet-of-things-definition/> (24.08.2021.)
- [5] <https://medium.com/zeux/the-internet-of-things-iot-5-reasons-why-the-world-needs-it-125fe71195cc> (25.08.2021.)
- [6] <https://www.redappletech.com/wp-content/uploads/2019/04/IoT-Smart-Home-Automation.png> (25.08.2021.)
- [7] <https://www.avsystem.com/blog/what-is-iot-architecture/> (04.09.2021.)
- [8] <https://www.iotforall.com/wifi-role-iot> (05.09.2021.)
- [9] <https://www.avsystem.com/blog/narrowband-iot/> (13.09.2021.)
- [10] [https://en.wikipedia.org/wiki/Narrowband\\_IoT](https://en.wikipedia.org/wiki/Narrowband_IoT) (13.09.2021.)
- [11] [https://static.wixstatic.com/media/bd334d\\_70e6350958fd4a11925d9867c9fb7a8b~mv2.jpg/v1/fill/w\\_640,h\\_266,al\\_c,lg\\_1,q\\_80/Proizvodi\\_NBPS\\_Cover.webp](https://static.wixstatic.com/media/bd334d_70e6350958fd4a11925d9867c9fb7a8b~mv2.jpg/v1/fill/w_640,h_266,al_c,lg_1,q_80/Proizvodi_NBPS_Cover.webp) (24.09.2021.)
- [12] <https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy> (24.09.2021.)
- [13] [https://en.wikipedia.org/wiki/Bluetooth\\_Low\\_Energy](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy) (24.09.2021.)
- [14] [https://blog.beaconstac.com/wp-content/uploads/2018/07/proximity\\_marketing\\_work.jpg](https://blog.beaconstac.com/wp-content/uploads/2018/07/proximity_marketing_work.jpg) (24.09.2021.)
- [15] <https://en.wikipedia.org/wiki/LoRa> (22.09.2021.)
- [16] [https://www.bosch-presse.de/pressportal/de/media/dam\\_images/pi10957/3.jpg](https://www.bosch-presse.de/pressportal/de/media/dam_images/pi10957/3.jpg) (07.09.2021.)
- [17] <https://www.sparkfun.com/products/14193> (07.09.2021.)
- [18] [https://cdn.sparkfun.com/assets/learn\\_tutorials/1/4/3/CCS811\\_Datasheet-DS000459.pdf](https://cdn.sparkfun.com/assets/learn_tutorials/1/4/3/CCS811_Datasheet-DS000459.pdf) (07.09.2021.)
- [19] <https://en.wikipedia.org/wiki/I%C2%B2C> (26.09.2021.)
- [20] <https://github.com/maarten-pennings/CCS811> (26.09.2021.)
- [21] [https://www.distrelec.biz/Web/WebShopImages/landscape\\_large/2-/01/Adafruit-2652-30091192-01.jpg](https://www.distrelec.biz/Web/WebShopImages/landscape_large/2-/01/Adafruit-2652-30091192-01.jpg) (07.09.2021.)
- [22] <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/> (04.09.2021.)
- [23] <https://lastminuteengineers.com/bme280-arduino-tutorial/> (04.09.2021.)
- [24] <https://en.wikipedia.org/wiki/ESP8266> (05.09.2021.)

- [25] <https://www.espressif.com/en/products/socs/esp8266> (05.09.2021.)
- [26] <https://www.auscomtech.com.au/wp-content/uploads/2018/12/AD483-2.jpg>  
(07.09.2021.)
- [27] <https://i.pinimg.com/originals/0c/8c/e8/0c8ce8bc22ee6ba0e2ed830bcbb28454.png>  
(13.09.2021.)
- [28] [https://www.wemos.cc/en/latest/d1/d1\\_mini.html](https://www.wemos.cc/en/latest/d1/d1_mini.html) (13.09.2021.)
- [29] <https://cdn2.peterdavehello.org/wp-content/uploads/2016/02/Raspberry-Pi-3-Model-B-Diagram-from-RS.png> (25.09.2021.)
- [30] <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (23.09.2021.)
- [31] <https://www.bigmessowires.com/wp-content/uploads/2018/05/Raspberry-GPIO.jpg>  
(23.09.2021.)
- [32] <https://www.u-blox.com/en/blogs/insights/mqtt-beginners-guide> (23.09.2021.)
- [33] [https://oringnet.com/upload/media/Technology/MQTT/MQTT\\_1091217-01.jpg](https://oringnet.com/upload/media/Technology/MQTT/MQTT_1091217-01.jpg)  
(24.09.2021.)
- [34] <https://selfhostedhome.com/setting-up-mqtt-broker-for-diy-home-assistant-sensors/>  
(26.09.2021.)
- [35] <https://pimylifeup.com/raspberry-pi-mosquitto-mqtt-server/> (26.09.2021.)

## 12. Popis slika

|   |    |
|---|----|
| Slika 1 Širok raspon uređaja spojenih na IoT <sup>[2]</sup> .....                 | 2  |
| Slika 2 Automatizacija doma pomoću IoT-a sve je češća <sup>[6]</sup> .....        | 4  |
| Slika 3 Četiri sloja IoT arhitekture <sup>[7]</sup> .....                         | 5  |
| Slika 4 Prikaz prvog sloja IoT arhitekture <sup>[7]</sup> .....                   | 6  |
| Slika 5 Prikaz drugog sloja IoT arhitekture <sup>[7]</sup> .....                  | 7  |
| Slika 6 Prikaz trećeg sloja IoT arhitekture <sup>[7]</sup> .....                  | 8  |
| Slika 7 Prikaz četvrtog sloja IoT arhitekture <sup>[7]</sup> .....                | 8  |
| Slika 8 NB-IoT parkirni senzor varaždinske tvrtke Mobilisis <sup>[11]</sup> ..... | 10 |
| Slika 9 BLE odašiljači (eng. <i>beacons</i> ) <sup>[14]</sup> .....               | 11 |
| Slika 10 Bosch LoRaWAN parkirni senzori <sup>[16]</sup> .....                     | 12 |
| Slika 11 CCS811 senzor <sup>[17]</sup> .....                                      | 13 |
| Slika 12 BME280 senzor <sup>[21]</sup> .....                                      | 16 |
| Slika 13 ESP8266 – 12E <sup>[26]</sup> .....                                      | 18 |
| Slika 14 ESP-12E raspored pinova <sup>[27]</sup> .....                            | 19 |
| Slika 15 Wemos (Lolin) D1 Mini razvojna pločica <sup>[28]</sup> .....             | 20 |
| Slika 16 Shema spajanja Wemos D1 i senzora u programu „Fritzing“ .....            | 20 |
| Slika 17 Wemos D1 i senzori zalemljeni na bušenoj pločici .....                   | 21 |
| Slika 18 Konektori na Raspberry Pi 3B modelu <sup>[29]</sup> .....                | 22 |
| Slika 19 Raspored GPIO pinova na Raspberry Pi pločici <sup>[31]</sup> .....       | 23 |
| Slika 20 Raspberry Pi 3B u kućištu .....  | 23 |
| Slika 21 Princip rada MQTT protokola <sup>[33]</sup> .....                        | 24 |
| Slika 22 Web aplikacija otvorena na pametnom mobitelu .....                       | 35 |

### 13. Popis tablica

|   |    |
|---|----|
| Tablica 1 Karakteristike CCS811 senzora <sup>[18]</sup> .....   | 14 |
| Tablica 2 Karakteristike BME280 senzora <sup>[22]</sup> .....   | 16 |
| Tablica 3 Vrijednosti koncentracije CO <sub>2</sub> i TVOC-a i utjecaj na ljude <sup>[18]</sup> ..... | 36 |

## 14. Popis grafova

|  |    |
|--|----|
| Grafikon 1 Izmjerena kvaliteta zraka u uredskoj prostoriji ..... | 36 |
| Grafikon 2 Izmjerena kvaliteta zraka u podzemnoj garaži .....    | 37 |
| Grafikon 3 Izmjerena kvaliteta zraka u hali za zavarivanje ..... | 38 |



## **Prilozi**

Prilog 1: CD-R disk