

Distribuirani sustav mjerenja okolišnih parametara temeljen na IoT

Puntarić, 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:200:119116>

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

Download date / Datum preuzimanja: **2024-12-27**

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

DISTRIBUIRANI SUSTAV MJERENJA OKOLIŠNIH

PARAMETARA TEMELJEN NA IOT

Završni rad

Marko Puntarić

Osijek, 2020.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 24.09.2020.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na
preddiplomskom sveučilišnom studiju**

| | |
|---|---|
| Ime i prezime studenta: | Marko Puntarić |
| Studij, smjer: | Preddiplomski sveučilišni studij Računarstvo |
| Mat. br. studenta, godina upisa: | R4123, 25.09.2019. |
| OIB studenta: | 69972252655 |
| Mentor: | Izv.prof.dr.sc. Tomislav Keser |
| Sumentor: | |
| Sumentor iz tvrtke: | |
| Naslov završnog rada: | Distribuirani sustav mjerenja okolišnih parametara temeljen na IoT |
| Znanstvena grana rada: | Procesno računarstvo (zn. polje računarstvo) |
| Predložena ocjena završnog rada: | Izvrstan (5) |
| Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova: | Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina |
| Datum prijedloga ocjene mentora: | 24.09.2020. |
| Datum potvrde ocjene Odbora: | 30.09.2020. |
| Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija: | Potpis: |
| | Datum: |

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 30.09.2020.

Ime i prezime studenta:

Marko Puntarić

Studij:

Prediplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R4123, 25.09.2019.

Turnitin podudaranje [%]:

11

Ovom izjavom izjavljujem da je rad pod nazivom: **Distribuirani sustav mjerenja okolišnih parametara temeljen na IoT**

izrađen pod vodstvom mentora Izv.prof.dr.sc. Tomislav Keser

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ:

| | |
|--|----|
| 1. UVOD | 1 |
| 1.1. Zadatak i ciljevi rada | 2 |
| 2. TEORIJSKI OSVRT NA PROBLEM MJERENJA OKOLIŠNIH PARAMETARA I PREZENTACIJE KORISNIKU I PRIJEDLOG RJEŠENJA | 3 |
| 2.1. Mjereni okolišni parametri i IoT platforma | 3 |
| 2.1.1. Teorija fizikalnog principa mjerenja parametara | 3 |
| 2.1.2. Teorija distribuiranih sustava | 4 |
| 2.1.3. Teorija integracija podataka | 4 |
| 2.1.4. Teorija mjerenja temperature, vlage, tlaka i koncentracije plinova | 5 |
| 2.1.5. Metode i postupci mjerenja temperature, tlaka i koncentracije plinova | 6 |
| 2.1.6. Teorija mjerenja buke | 7 |
| 2.1.7. Metode i postupci mjerenja buke | 7 |
| 2.1.8. Teorija mjerenja navedenih svjetlosnih parametara | 8 |
| 2.1.9. Metode i postupci mjerenja svjetlosnih parametara | 10 |
| 2.1.10. Teorija vizualnog prikaza čovjeku prihvatljivog oblika prezentacije podataka | 10 |
| 2.2. Idejno rješenje | 11 |
| 2.3. Prijedlog algoritamskog rješenja | 12 |
| 2.4. Prijedlog upravljanja i sučelja čovjek-stroj | 12 |
| 3. REALIZACIJA SUSTAVA | 13 |
| 3.1. Korišteni alati i razvojna okruženja | 13 |
| 3.1.1. Arduino NANO mikrokontroler | 13 |
| 3.1.2. Raspberry Pi 3 Model B+ | 14 |
| 3.1.3. BME280 | 14 |
| 3.1.4. BH1750 | 15 |
| 3.1.5. LM393 mikrofonski modul | 16 |
| 3.1.6. MQ135 senzor općenite kvalitete zraka | 16 |
| 3.1.7. NRF24L01+ 2.4Ghz radio modul | 17 |
| 3.1.8. 7805 5V REGULATOR NAPONA THT | 18 |
| 3.1.9. Arduino IDE | 18 |
| 3.1.10. Eagle | 19 |
| 3.2. Sklopovska realizacija | 19 |
| 3.3. Algoritamska realizacija | 23 |
| 3.4. Sučelje čovjek-stroj i komunikacijska infrastruktura | 26 |
| 4. TESTIRANJE I REZULTATI | 28 |

| | |
|--|-----------|
| 4.1. Metode i načini testiranja | 28 |
| 4.2. Rezultati testiranja | 28 |
| 5. ZAKLJUČAK..... | 37 |
| LITERATURA | 38 |
| PRILOZI..... | 39 |
| POPIS SLIKA..... | 44 |

1. UVOD

S više od desetljeća intenzivnog istraživanja i razvoja, pojavila se tehnologija bežične senzorske mreže kao održivo rješenje mnogih inovacija. Velik broj bežičnih mrežnih sustava senzora razvijeni su pomoću hardverskih platformi otvorenog koda, kao što su Arduino, Raspberry Pi i platforme Internet stvari (engl. Internet of things, IoT). Ti sustavi su jeftini i vrlo skalabilni, kako u pogledu vrste senzora, tako i broja senzorskih čvorova, što ih čini pogodnim za široku paletu aplikacija povezanih s nadzorom okoliša. Primjer jedne moguće arhitekture sustava i dizajn hardverskih i softverskih komponenti predstavljeni su u ovom radu.

Raniji radovi na senzorskim mrežama i fizičkim sustavima bili su usmjereni na razvoj mogućih tehnologija rješavanjem bezbroj tehničkih izazova kao višestruko usmjeravanje, komunikacijske apstrakcije, srednji softver i operativni sustavi (OS), semantičke apstrakcije, dijeljenje podataka te usavršavanje senzora. Većina ranih testnih sustava izgrađena je korištenjem istraživanja mreže senzora u ranoj fazi platforme poput CrossBow-a (sada MEMSIC) i TinyOS-ovog softverskog okvira. Mreže senzora su u osnovi ugrađeni mikrokontrolerski sustavi male potrošnje s raznim ugrađenim sensorima i analognim ili digitalnim ulazno/izlaznim priključcima za spajanje senzora. Također potrebno je razviti paket softverskih komponenti, ponekad uključujući OS, upravljačke programe senzora / hardvera, protokole za umrežavanje i algoritme osjetljivosti i obrade specifične za određenu potrebu. Postoji veliki broj dosadašnjih napora koji su pronašli uspješnu primjenu. Razvijen je mrežni sustav detekcije vlage tla velikih razmjera pomoću CrossBow-a i integriran je u sveobuhvatni sistem zaštite okoliša, Texas Environmental Observatory (TEO)[1]. Nadalje, MoteLab bila je eksperimentalna bežična senzorska mreža uspostavljena na Sveučilištu Harvard. Pružila je javno dostupno testiranje za razvoj i testiranje senzorskih mrežnih aplikacija putem intuitivnog internetskog sučelja. Otpisana je u siječnju 2014. Još jedan primjer je INDRIYA, razvijena trodimenzionalna bežična senzorska mreža na Nacionalnom sveučilištu u Singapuru, bazira se na istom dizajnu kao i MoteLab[2].

U novije vrijeme svjedoci smo novog vala razvoja na području hardvera / softvera otvorenog koda, standardizacije, i širokoj primjeni bežičnih senzorskih mrežnih tehnologija. IEEE 802.15.4 standard je jedna od prednosti koja olakšava postavljanje lokalne mreže s velikom brzinom prijenosa pogodnom baš za ovakve potrebe. U kontekstu IoT-a, važni su protokoli sloja linka: IEEE 802.3 Ethernet, 802.11 WiFi, 802.16 WiMax, navedeni 802.15.4 LR-WPAN (kojeg koristi npr. ZigBee tehnologija).

Također i komunikacijski protokoli mobilnih tehnologija 2G (GSM, CDMA), 3G,4G a već i 5G. Važna tehnologija sloja linka je i LoRaWAN, bluetooth.

Sustavi senzorskih mreža, poput većine ugrađenih sustava, trebaju biti usko povezani sa njihovim primjenama. Međutim, spomenuta dostignuća pomogla su smanjenju složenosti primjene bežičnih senzora i pokretačkih sustava i omogućila su prilično jednostavnu implementaciju sustava.

U ovom radu predstavljen je bežični IoT distribuirani sustav mjerenja okolišnih parametara uz pomoć senzora. Razvijen je korištenjem platformi otvorenog koda, Arduino i Raspberry Pi, mjernih senzorskih uređaja te modula za komunikaciju između navedenih.

Takav dizajn ima prednost niske cijene, izrazito lako dodavanje novih funkcija rada i reprogramiranje, te jednostavnosti za izgradnju i održavanja u usporedbi s nekim ranijim dizajnima, poput TEO sustava. Glavne prepreke leže u nedostatku pouzdanosti, fleksibilnosti i u nadolazećim teškoćama u dugoročno razvijanje, rad i održavanje, posebno od strane krajnjih korisnika koji nisu inženjeri.

1.1. Zadatak i ciljevi rada

Zadatak završnog rada je dizajnirati i izraditi sustav koji pomoću nekoliko mjernih čvorova prikuplja željene parametre iz okoliša. Primijenit će se tehnologija Interneta stvari (engl. Internet of things, IoT) kroz scenarij komuniciranja udaljenih senzora sa centralnom jedinicom koja parametre prikuplja, obrađuje, pohranjuje i prikazuje putem prikladne IoT platforme koja će predstavljati sučelje koje će omogućiti krajnjem korisniku pristup mjernim podacima.

2. TEORIJSKI OSVRT NA PROBLEM MJERENJA OKOLIŠNIH PARAMETARA I PREZENTACIJE KORISNIKU I PRIJEDLOG RJEŠENJA

Okoliš je sve što nas okružuje, čini nam se nepoznatim i složenim, no ako određene parametre izdvojimo i mjerimo, te prikazemo pomoću dostupne tehnologije i poznatih veličina okolina postaje jednostavnija i shvatljivija.

Sustav mjerenja je skup umreženih senzorskih jedinica povezanih na mikrokontrolersku centralnu jedinicu kojemu je svrha prikupljanje podataka o kakvoći zraka, okolišnog osvjetljenja, buke, temperature i koncentraciji određenih plinova.

Mjerenje je proces koji zahtjeva konkretne postupke i testiranja, stoga je cilj mjerenja dobiti čitljive vrijednosti mjerenog parametra. Mjerenjem istih možemo procijeniti u kakvim uvjetima se nalazimo i kakvo je stanje u okolini, te parametre korisno je prikazati nama poznatim veličinama, te na temelju toga istraživati i korigirati trenutne uvjete ako je to potrebno.

2.1. Mjereni okolišni parametri i IoT platforma

Izabrani su parametri koji su mjereni u ovom radu te će oni biti teorijski detaljnije objašnjeni. U radu se lokalno mjere parametri okolišnog zraka. To su: temperatura, relativna vlaga, tlak i koncentracija određenih plinova. Zatim svjetlosne značajke kao: jakost osvjetljenja i intenzitet zračenja. Prati se i intenzitet buke u čovjekovom čujnom dijelu spektra.

2.1.1. Teorija fizikalnog principa mjerenja parametara

Kako bi izmjerili željene parametre potrebni su nam gotovi senzor moduli ili komponente čija očitavanja možemo iskoristiti te pomoću njih dobiti vrijednost mjerenog parametra. Sensori ne mogu sami po sebi očitati neki parametar iz zraka, ne postoji određena specifična komponenta koja reagira na temperaturu zraka ili na razinu vlage i slično. Neki koriste diode, infracrvene senzore, neki otpornike, te komponente reagiraju na uvjete u okolišu te dodatnim izračunom daju povratnu informaciju. Zajedničko im je to da neku fizičku veličinu pretvaraju u električnu, koju unutarnjim električnim sklopom prvo obrađuju kao analognu, potom je pretvaraju u digitalnu, eventualno propuštaju kroz digitalni filter i linearizator, a onda ju čine dostupnom preko porta za komunikaciju sa mikrokontrolerom.

Razvijanjem IoT područja tehnologije, senzori dobivaju veću pažnju jer omogućuju da se mjereni parametar približno trenutno pošalje na bilo koje udaljeno mjesto. To je rezultiralo naglim razvojem, dostupnosti širokoj populaciji i izradom brojnih projekata.

2.1.2. Teorija distribuiranih sustava

Prihvaćena definicija distribuiranih sustava objašnjava kako je distribuirani sustav skup neovisnih članova, poput mikrokontrolera, korištenog hardvera i softverske opreme koji zajedno za krajnjeg korisnika predstavljaju jedan cjeloviti sustav. Hardverski dio definicije bi značilo više nezavisnih članova spojenih u mrežu, dok softverski dio obuhvaća sustav koji omogućuje međusobnu koordinaciju svih potrebnih aktivnosti između članova.

Ono što se pokušava postići u ovom radu kada dizajniramo i izrađujemo distribuirani sustav je povezanost, samostalnost i prikladan dizajn sustava, učinkovitu komunikaciju i funkcionalnost te što manje pogreške. Ključne osobine distribuiranih sustava čine hardverske i softverske sastavnice koje se mogu primijeniti u različitim zadacima čvorova, te komuniciranje i zajednički rad komponenti.

U ovom radu prepreke će biti dizajniranje hijerarhije sustava, postizanje ravnopravnost senzorskih čvorova, kako ih razlikovati i identificirati, te kako postići usklađenu komunikaciju.

2.1.3. Teorija integracija podataka

Svaki informacijski sustav temelji se na podacima koje sadrži, koji se u njega unose i kroz njega održavaju. U praksi imamo sve više različitih aplikacija za prikupljanje i analizu podataka, a podatke iz njih je nužno prikupljati, razmjenjivati i usklađivati kako bi se osigurala pravovremena informacija za korisnike. Problem integracije podataka iz različitih sustava, te razmjene podataka među sustavima već je dugo prisutan, na tome se radi, uvode se standardi, a s vremenom dobiva sve više na važnosti. Zahtjev je postići da se podaci iz različitih sustava, koji za komunikaciju koriste različite standarde i protokole, iz različitih aplikacija na različitim jezicima, razmjenjuju unutar i među sustava.

Prilikom dizajniranja i programiranja mjernih čvorova i centralne jedinice potrebno je voditi računa o veličini podataka, formatu podataka koji se šalju, sinkroniziranju vremena slanja, ispravnom primanju podataka te usklađenom radu cijele dizajnirane mreže. Postoje alati koji nam mogu pomoći u tome, to su biblioteke koje pružaju gotove funkcije za korištenje koje rade u pozadini na razinama koje programer ne vidi i ne mora voditi brigu o tome.

2.1.4. Teorija mjerenja temperature, vlage, tlaka i koncentracije plinova

Senzori **temperature** su već dugo u upotrebi, ali nikada točniji, efikasniji i jednostavniji za uporabu kao danas. Najčešće su se koristili NTC otpornici (negativni temperaturni koeficijent) ili PTC (pozitivni), koji su vrlo nelinearni, te im je za rad potrebna dodatna konfiguracija. To je dodatno kompliciralo uporabu jer je tražilo analogni sklop, ali i pouzdan termometar te termičku komoru proces konfiguracije kako bi se postigla tražena točnost. Sadašnji senzori temperature dolaze integrirani sa gotovim električnim sklopom i jedinicom za komunikaciju, većinom za I2C model. Pri mijenjanju temperature mijenja se i otpor. Senzor očitava vrijednost i pretvara ga u digitalni signal. Još jedan problem koji se pojavljivao kod starijih senzora je zagrijavanje kućišta, a i samoga senzora, zbog same potrebne elektronike i zatvorenog kućišta, tako da je to moglo rezultirati pogrešnim rezultatima. Suvremeni senzori imaju malu potrošnju energije, pa je pogreška svedena na minimum. Termički masivniji senzor osigurati će točnije mjerenje, ali će sporije odgovoriti na izmjene temperature mjerenog okoliša. Zadovoljavajući odgovor su senzori koji su manji i termički neovisni, te se pale samo ponekad, idealno kada i očekivane izmjene temperature.

Sadašnji senzori za mjerenje relativne **vlažnosti** zraka često dolaze zajedno sa sensorima temperature i zajedno se nalaze u kućištu ili na sklopu vrlo malih dimenzija, usporedivo sa veličinom kovanice. Pojedini senzori znaju imati otvorene vodove koji su međusobno vrlo blizu. Što je više vlage u zraku, više dolazi i na same vodove te je otpor među njima manji. Poseban problem u održavanju ovih senzora nastaje zbog osjetljive prednje strane senzora koja propušta zrak do mjerne površine, pa ih nije preporučljivo izložiti velikoj vlažnosti. Senzori vlage tada neko vrijeme „pamti“ ekstremnu vrijednost. Zato se poslije svake nagle promjene preporučuje neko vrijeme rada u uvjetima srednje relativne vlažnosti zraka, i tek poslije rezultat očitavanja može se smatrati preciznim.

Senzori **tlaka** baziraju se na principu pretvaranja tlaka zraka u mehaničko kretanje. Senzor tlaka čine pretvornik s osjetljivim elementom, kućište, sami mehanički elementi (opruga i dijafragma) i sklop.

Senzor za **koncentraciju određenih plinova**, radi tako da s povećanjem koncentracije plina raste i izlazni napon koji se kasnije odgovarajućom skalom pretvara u udio plina i dobivamo željenu očitane vrijednost. Pozicije za različite plinove je bitno točno odrediti jer krivim vrijednostima senzori neće detektirati prisutnost plina, što može rezultirati daljnjim komplikacijama. Senzoru je potrebno vrijeme zagrijavanja i kalibracija kako bi ispravno radio.

2.1.5. Metode i postupci mjerenja temperature, tlaka i koncentracije plinova

Kako bi se izmjerila **temperatura** može se odabrati senzor otpornikom, termistor ili termoparovi. Kod senzora otpornikom postupak rada temelji se na promjeni otpora, koji je ovisan temperaturi. Struja teče kroz senzor i uzrokuje pad napona koji je također zavisi o otporu, dakle o temperaturi. Termistori su otpornici osjetljivi na temperaturu kojima se električni otpor raste ili pada ovisno o temperaturi. Dijelimo ih na dvije vrste: NTC (negativni temperaturni koeficijent), to su termistori koji se koriste za mjerenje temperature i PTC (pozitivni temperaturni koeficijent), to su termistori koji se nalaze u sigurnosnim sklopovima. NTC vrsti otpor se smanjuje s rastom temperature. Termoparovi kao treća vrsta su senzori koji čine dvije žice različitih materijala spojene na jednom zajedničkom kraju. Postupak rada termopara temelji se na termoelektričnom efektu. Pri izlaganju metala temperaturnom gradijentu dolazi do pojave napona.

Određeni senzori **temperature** su vrlo precizni integrirani krugovi s naponom proporcionalnim s temperaturom koja je izražena u Celzijusu. Nisu potrebne dodatne kalibracije kako bi dali točnost od $\pm 0,5$ °C pri umjerenom temperaturi.

Vlažnost zraka može se prikazati na više načina, a često korišteni je relativna vlažnost. Relativna vlažnost zraka pri određenoj temperaturi jednaka je omjeru tlaka vodene pare i tlaka zasićene vodene pare pri istoj temperaturi. Jedan od mogućih postupaka mjerenja vlage zraka je koristeći kapacitivni senzor. Takav senzor čini podloga na kojoj je tanki film polimera ili metalnog oksida smješten među dvije vodljive elektrode. Osjetilna površina obavijena je poroznim materijalom kako bi ju osigurao od zagađenja i kondenzacije. Podloga je najčešće staklena, može biti od keramike ili čak silicija.

Spomenuti sloj polimera je dielektrik koji prima molekule vode iz zraka, pri čemu se izmjenjuju dielektrična svojstva, a zatim i kapacitet. Kapacitet je proporcionalan relativnoj vlažnosti. Ovakvi senzori vlažnost prikazuju u rasponu mjerenja od 1 do 100% s pogreškom od $\pm 3\%$. [3]

Atmosferski tlak je tlak na bilo kojem dijelu Zemljine atmosfere. Jedinica za tlak je Paskal (Pa), dok se u češće koristi hektopaskal (hPa). Jedan hektopaskal jednak je jednom milibaru (1hPa = 1mbar). Tlak zraka ovisi o visini, pa se pri izražavanju tlaka zraka on svodi na srednju razinu mora kako bi se mogao usporediti sa ostalim mjernim točkama.

Jedan od načina mjerenja tlaka je mjerenjem pomaka osjetnog elementa. Zbog razlike tlakova doći će do pomaka membrane, mijeha, ili spiralna cijevi, što se može izmjeriti pretvornikom pomaka. U automatiziranim mjernim stanicama koriste se senzori kod kojih se mjerenje bazira na mjerenju promjene napona te promjene otpora koja nastaje kao rezultat naprezanja u osjetnom elementu. [4]

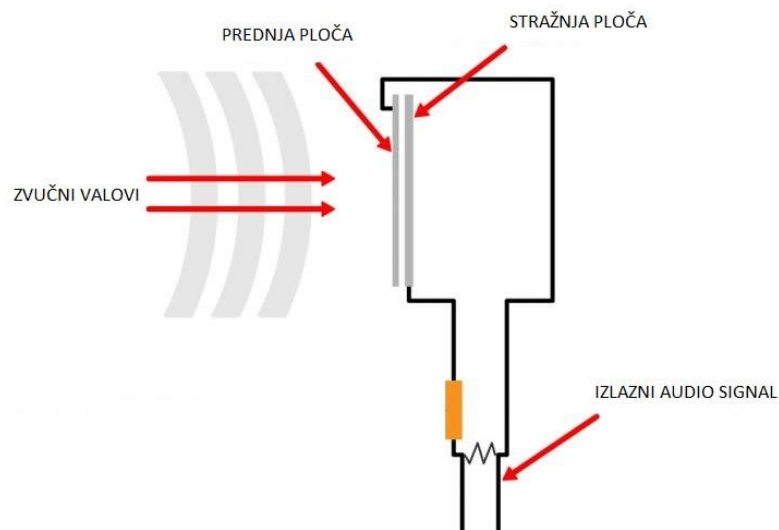
2.1.6. Teorija mjerenja buke

Sastavni dio često korištenih modula za mjerenje zvuka je **kondenzatorski mikrofون** koji energiju zvuka pretvara u električnu. Takvi mikrofoni izrađeni su od membrane i čvrste ploče. Titranjem membrane varira razmak između elektrode, što rezultira promjenom kapaciteta, odnosno napona. Kada nema očitanih zvukova izlazna logika modula je HIGH i suprotno LOW. Neki moduli za detekciju zvuka sadrže signalnu led lampicu koja se pali kada modul detektira zvuk te potencijometar za postavljanje granične vrijednosti.

Na modulu se nalazi i analogni komparator napona. On mjeri napon u dvije različite točke te ih uspoređuje. Kada je napon u prvoj točki veći od napona u drugoj točki sklop je upaljen, u protivnom stanju je isključen. Granicu za usporedbu postavljamo pomoću montiranog potencijometra na modulu. To omogućuje da se modul koristi i bez dodatnih kontrolera.

2.1.7. Metode i postupci mjerenja buke

Kondenzatorski mikrofون sastoji se od dvije električki nabijene ploče. Jedna od njih može se pomicati i ponaša se kao membrana, a druga je fiksna. To ustvari predstavlja kondenzator sa negativno i pozitivno nabijenim elektrodama i zračni prostor između elektroda. Primljeni zvuk pomiče membranu, što uzrokuje promjenu razmaka između nje i fiksne ploče. Stoga ta promjena mijenja kapacitet i razmak između membrane i statične ploče te dovodi do promjene napona koji je potrebno pojačati do iskoristive razine. Pojačanje navedenoga vršimo pojačalima.

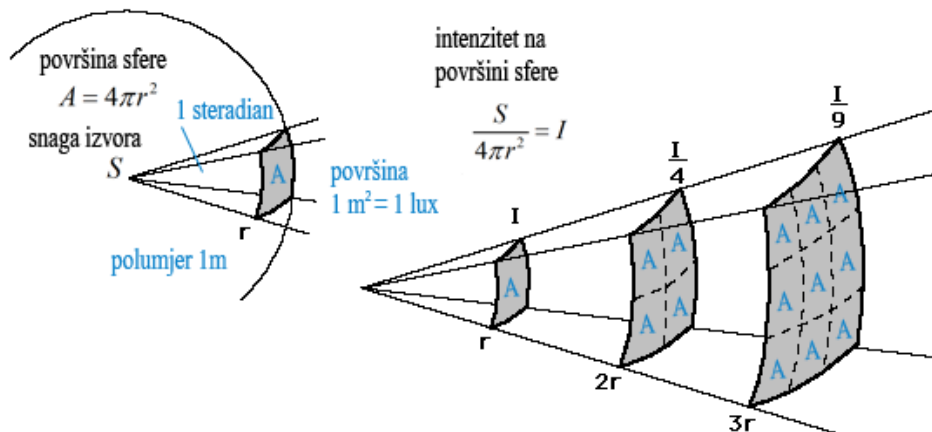


Slika 2.1. komponente kondenzatorskog mikrofona

2.1.8. Teorija mjerenja navedenih svjetlosnih parametara

Mjerenje osvjetljenja u okolišu izvodi se pomoću **foto otpornika ili foto diode**. Foto otpornik je komponenta s promjenjivim otporom zavisnom o količini svjetla. Česta primjena je detekcija svjetla ili mraka. Postoji širok spektar foto otpornika, a otpornosti se kreću od nekoliko Ω do nekoliko $M\Omega$. Postupak rada je jednostavan, otpor izložen svjetlosti se smanjuje i to uzrokuje da mjereni napon raste, kada se razina svjetla smanji njegov otpor raste što uzrokuje smanjenje mjenog napona.

Valna duljina je osnovna karakteristika svjetlosne zrake. Svjetlo putuje zrakom ponašajući se kao val, a razlika između dva susjedna vrha vala nazivamo valna duljina (λ - lambda). Jednostavnije sročeno, valna duljina određuje boju svjetlosne zrake. Intenzitet je druga karakteristika svjetlosne zrake. Mjerna jedinica SI sustava je Watt po Steradianu (W / sr).



Slika 2.2. Intenzitet svjetla na površini sfere

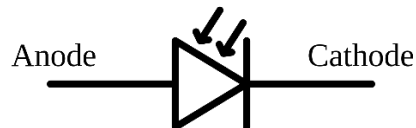
Prema formuli sa slike, površina dvostruko udaljenija od izvora pružit će dvostruko manji intenzitet svjetla.[5]

U modulu senzora postupak rada počinje od fotodiode kojoj je osjetljivost približno ljudskom oku. Dobiveni signal pojačava se pomoću operacijskog pojačala te se pretvara iz analognog u digitalni. Postupak svodimo kraju IIC ili I2C logikom koja koristi interni oscilator. Trajanje ovog postupka ovisi o točnosti očitavanja koju zahtjevamo. Zbog duljeg mjerenja može doći do eliminacije nekih šumova, stoga dolazimo do točnijih rezultata. Lx je SI mjerna jedinica osvjetljenja. Svakako treba naglasiti kako se rang mjerenja može povećati do 10k lx, što odgovara svjetlosti koju pruža direktna sunčeva svjetlost.

2.1.9. Metode i postupci mjerenja svjetlosnih parametara

Fotodioda je poluvodička komponenta strukture poput ispravljačke diode. Kućište ima otvor ili prozor kroz koji se osvjetljava PN-spoj. Pri izradi fotodioda koriste se germanij i selenij, a najviše silicij. Fotodioda se u strujni krug povezuje zaporno polarizirana. Kada fotodioda nije osvjetljena, kroz nju teče vrlo malena tamna struja koju čini reverzna struja (struja manjinskih nosioca naboja). Vrijednost te struje za silicijske fotodiode iznosi nekolicinu nanoampera, a za germanijske nekoliko mikroampera. Kada osvjetlimo fotodiodu, reverzna struja se iznimno poveća (poraste broj slobodnih nosioca naboja). Iznos tog porasta struje kod germanijskih dioda iznosi oko 20 puta, dok kod silicijskih 100-500 puta.

Fotoosjetljivost je odnos struje osvjetljene diode i osvjetljenost koja tu struju uzrokuje, a izražava se u lx.[6]



Slika 2.3. oznaka za fotodiodu

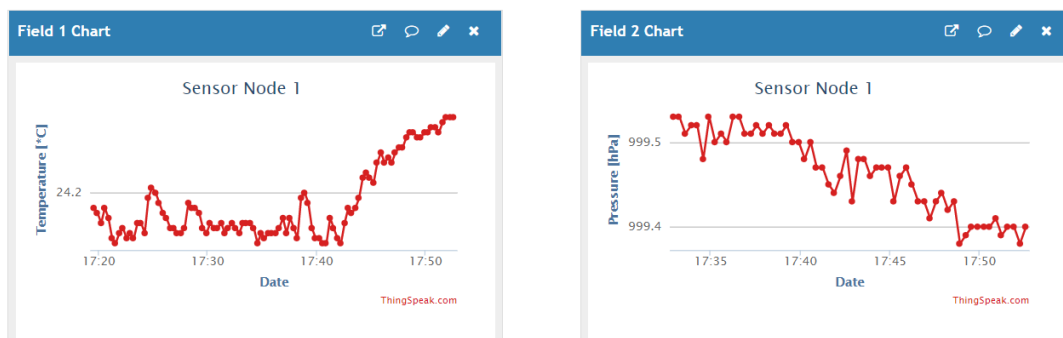
2.1.10. Teorija vizualnog prikaza čovjeku prihvatljivog oblika prezentacije podataka

Kao što je potrebno precizno i točno izmjeriti parametre i analizirati ih, jednako tako potrebno ih je i prikazati ih u lako razumljivom obliku. Podatci se nakon prikupljanja u centralnoj jedinici ili IoT platformi razvrstavaju, evaluiraju i postavljaju korisnicima na pregled i postaju dostupni za daljnje korištenje. Zavisno o formatu podataka možemo odrediti najprihvatljiviji oblik prezentacije, bio to graf u ovisnosti o nekim parametrima ili matematička funkcija praćenog parametra. Dostupna su brojna gotova rješenja poput ThingSpeak-a.

ThingSpeak je usluga platforme IoT za analitiku koja nam omogućuje okupljanje, vizualizaciju i analizu protoka podataka uživo u oblaku.

Channel Stats

Created: [26 days ago](#)
 Last entry: [3 days ago](#)
 Entries: 316



Slika 2.4. prikaz temperature (lijevo) i tlaka (desno) pomoću IoT ThingSpeak platforme

2.2. Idejno rješenje

Dizajnirat će se mreža mjernih/senzorskih „čvorova“ i centralne jedinice koji će međusobno ostvariti bežičnu komunikaciju između uređaja na IoT principu. Svaka senzorska jedinica bit će postavljena na nekoj udaljenosti od centralne jedinice. Imat će svoju bateriju koja će služiti kao nezavisno napajanje i regulator napona za Arduino razvojnu pločicu, senzorske module i modul za komunikaciju. Sve od navedenog nalazit će se u kućištu pri čijoj izradi treba voditi računa o zbijenosti komponenti, protoku zraka kako bi mjerenja bila što preciznija, zagrijavanju. Izbor Arduino razvojne ploče ovisi ponajprije o broju ulaza i izlaza potrebnih za sve senzore te kompaktnosti. Svaki čvor sastojat će se od 4 senzorska modula. Prvi za temperaturu, tlak i vlagu, drugi za svjetlosne parametre, treći modul za mjerenje intenziteta buke, te modul za koncentraciju određenih plinova u zraku. Senzorski čvor obavit će pred obrađivanje kako bi se pogreške mjerenja svele na minimum, te pripremu podataka u željenu formu koja se zatim šalje centralnoj jedinici.

Svaki mjerni čvor ostvarit će bežičnu komunikaciju sa centralnom jedinicom. To će biti ostvareno pomoću primopredajnog modula za bežičnu komunikaciju. Potrebno je programskim rješenjem osigurati da se podaci ne šalju u istom trenutku i bez reda kako ne bi došlo do miješanja i nemogućnosti primanja željenih podataka. Centralna jedinica dobivat će napajanje koristeći

AC-DC strujni adapter. Kao malo računalo zaduženo za obavljanje zadataka i funkcija centralne jedinice služiti će Raspberry Pi sa potrebnom periferijom.

Veličine je kreditne kartice a dovoljno snažan i sa svim mogućnostima potrebnim za ovu mrežu. Prikupljeni podaci će tada biti obrađeni i pohranjeni i prikazani putem prikladne WEB tehnologije. Tu će se pokazati integracija IoT sustava. Za programsku realizaciju Arduino razvojnih pločica i senzorskih modula koristit će se IDE, koje je razvijeno za Arduino. Programski jezik za programiranje Arduino razvojne pločice biti će C++ te Python za Raspberry Pi.

2.3. Prijedlog algoritamskog rješenja

Algoritamsko rješenje zamišljeno je na principu programiranja neovisnog rada 4 senzorska čvora i centralne jedinice. Senzorski čvorovi koristit će vrlo slične algoritme zbog slične prirode rada. Centralna jedinica doći će u dodir sa senzorskim čvorovima jedino u trenutku ostvarivanja komunikacije koji je zamišljen da se ponavlja u određenom vremenskom intervalu i trajat će vrlo kratko. Svaki senzorski čvor imat će nekoliko osnovnih zadataka za obaviti te će za svaku biti napisana funkcija. To su redom: inicijalizirati senzore, utvrditi da je sve u redu i da je spreman početi s mjerenjem, obaviti mjerenje uz pomoć spojenih senzora, podatke spremirati, prikladno obraditi i pripremiti za slanje, ostvariti komunikaciju sa centralnom jedinicom i poslati mjerene parametre, čekati određeno vrijeme te početi iznova od koraka mjerenja. Centralna jedinica podatke će primiti, obraditi, detektirati iz kojeg senzorskog čvora su primljeni, zatim ih postaviti na ThingSpeak IoT platformu te u pripravnosti čekati nove podatke senzorskih jedinica.

2.4. Prijedlog upravljanja i sučelja čovjek-stroj

Upravljanje sensorima od strane korisnika bit će svedeno na minimum. Potrebno ih je samo postaviti na željene lokacije i pružiti im napajanje. Centralnoj jedinici dodatno je potrebno pružiti stabilnu internetsku vezu kako bi podizanje podataka na web platformu bilo moguće. Raspberry Pi će također samostalno ostvariti vezu sa sensorima uz pomoć prethodno napisane Python skripte koja će se pokrenuti pri paljenju. Sučelje koje će koristiti kao prozor između korisnika i stroja biti će ThingSpeak IoT platforma. Ondje će korisnik biti u mogućnosti pregledati mjerene parametre u stvarnom vremenu sa vrlo malom odgodom sa bilo kojeg uređaja sa internet vezom. Broj korisnika koji mogu promatrati podatke nije ograničen, kanali senzora mogu biti postavljeni na javne postavke i biti dostupni svima. Svaki mjereni parametar imat će svoj graf i prikaz kako bi korisnik što jednostavnije pronašao ili analizirao podatke. Također, za bolju i detaljniju analizu podataka na ThingSpeak platformi korisniku je ponuđen i MatLab kao pomagalo.

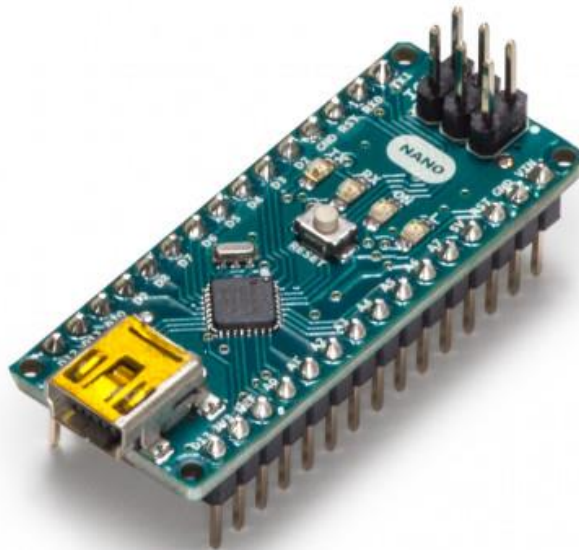
3. REALIZACIJA SUSTAVA

3.1. Korišteni alati i razvojna okruženja

3.1.1. Arduino NANO mikrokontroler

Arduino je skup mikrokontrolera zasnovanih na Atmel tehnologiji i pogodnih za razvoj upravljačke elektronike i robotike. Čitav sustav je otvorenog koda temeljen na jednostavnim razvojnim pločicama s ulaznim i izlaznim konektorima te besplatnom programskom podrškom s intuitivnim korisničkim sučeljem. Programiranje uređaja izvodi se uz pomoć integriranog razvojnog okruženja, dostupno je za Windows, Mac i Linux operacijske sustave, u programskom jeziku sličnom C-u, C++-u. Glavna uporaba cijelog sustava je komuniciranje s različitim komponentama koje su na njega priključene. Arduino je prvotno dizajniran i razvijen u Italiji za poučavanje programiranja u C/C++ programskim jezicima. Napravljen je od hardverskih komponenti koje su spojene na 8-bitni Atmel AVR mikrokontroler ili 32-bitni ARM procesor. Svi se modeli mogu proširivati tako zvanim štitovima. Modeli Arduino pločica između sotaloga razlikuju se po veličini, broju ulazno/izlaznih jedinica, radnom naponu, procesoru.

Arduino NANO je model sa sljedećom tehničkom specifikacijom: **P 3.1.**



Slika 3.1. Arduino Nano

3.1.2. Raspberry Pi 3 Model B+

Raspberry Pi je hardver računalne platforme centralne jedinice za pohranu i prezentaciju podataka. Linux operacijski sustav kojeg Raspberry Pi 3 koristi je Raspbian. To je besplatan sustav otvorenoga koda instaliran na Micro-SD memorijsku karticu priključenu u Raspberry Pi, odnosno memoriju pohrane s koje Raspberry Pi učitava operacijski sustav.



Slika 3.2. Raspberry Pi 3 model B+

Raspberry Pi 3 karakteriziraju male dimenzije ali široke mogućnosti primjene zbog gotovo identične funkcionalnosti, no nešto slabijih performansi, kao kod računala većih dimenzija te nalazi primjene u industriji zbog 40 konektorskog sučelja i 24 ulazno/izlazna konektora opće namjene. Tehničke karakteristike: **P 3.2.**

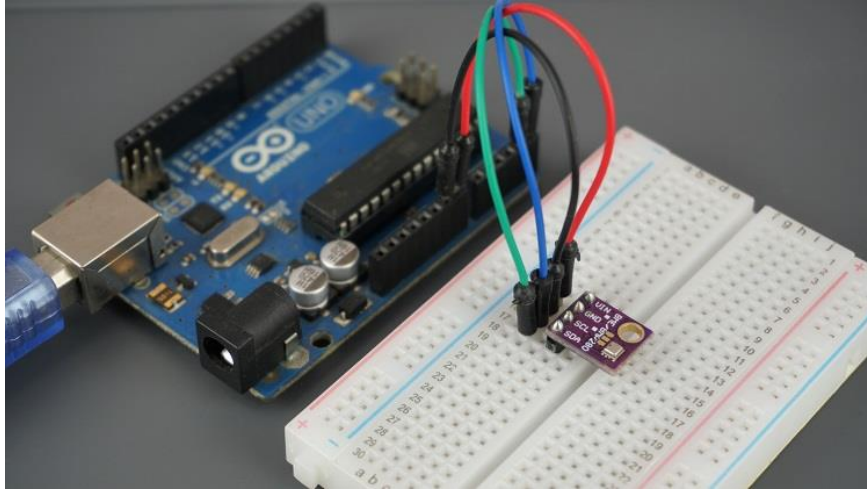
3.1.3. BME280

BME280 senzor temperature, tlaka i vlage dizajnira i proizvodi široko poznata tvrtka Bosch. Idealan je za razne projekte koji na bilo kakav način uključuju rad s vrijednostima navedenih vremenskih uvjeta, od vremenske stanice, motrenja okoliša, upravljanja grijanjem/hlađenjem, ali i druge situacije kao na primjer u pametnim satovima.

BME280 može biti postavljen u 3 različita načina rada: normal, sleep, forced:

Sleep mode karakterizira najmanja potrošnja, ne obavlja se mjerenje, i svi registri su spremni za očitavanje. Forced mode je korišten za situacije u kojima nije nužno napraviti puno čestih mjerenja, na primjer kod očitavanja vremenskih uvjeta. Obavi se mjerenje, vrijednosti se spremne u registre i senzor se postavlja u sleep mod. Na taj način smanjit će se potrošnja.

Normal mode konstantno obavlja mjerenja uz navedeni vremenski razmak, odnosno neaktivno vrijeme koje možemo podesiti.[7] Kako bi ga programirali u Arduino IDE-u moramo koristiti biblioteku BME280.h.



Slika 3.3. BME280 modul povezan na Arduino Uno

Određeni detalji su povjerljivi, ali BME280 koristi sljedeća načela mjerenja:

Temperatura: napon diode

Pritisak: otpor

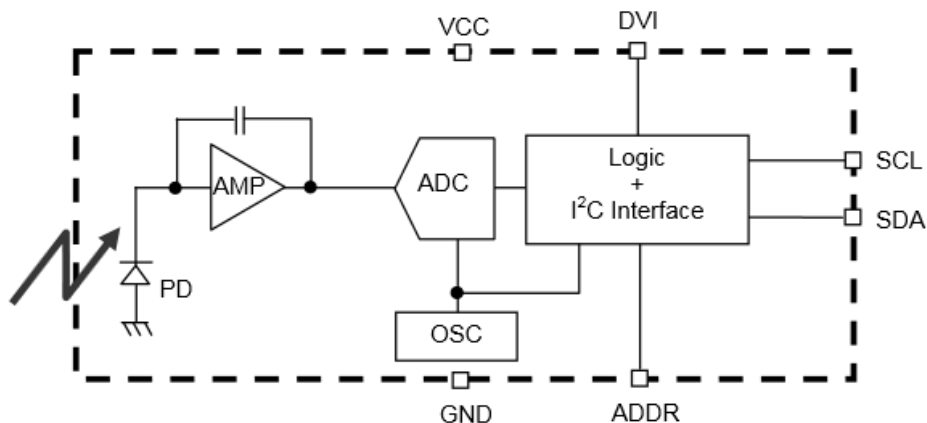
Vlaga: kapacitet

Stvarni princip mjerenja iza senzora zapravo ne diktira performanse.

3.1.4. BH1750

BH1750 je digitalni senzor svjetlosti opremljen sa I2C komunikacijom. Modul ima sve potrebno za komunikaciju s Arduinoom te integrirani regulator napona. Iznimno je stabilan te ima vrlo mala odstupanja s obzirom na pruženi izvor svjetlosti te izmjenu temperature.

Kako bi ga programirali i lakše koristili u Arduino IDE, poželjno je koristiti biblioteku BH1750.h.



Slika 3.4. Blok dijagram IC-a (Fotodioda(PD), Operacijsko pojačalo (AMP))

3.1.5. LM393 mikrofon modul

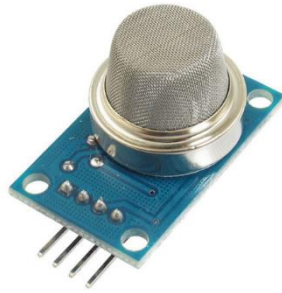
Ovaj modul mikrofona prepoznat će dolazne audio signale i pretvoriti ih u analogni ili digitalni izlaz prema mikrokontroleru na koji je povezan. Koristi LM393 čip za dodatno pojačavanje primljenog audio signala prije nego što ga pošalje na spojeni mikrokontroler.



Slika 3.5 LM393 zvučni senzor modul

3.1.6. MQ135 senzor općenite kvalitete zraka

MQ135 je senzor opće kvalitete zraka, koja se izražava veličinom ppm (eng.: parts per million). Senzor detektira plinove kao što su amonijak, sulfidi, benzen, dim. Češće se koristi za mjerenje same kvalitete zraka. Senzoru je za rad potreban napon od 5 V. Senzor sadrži analogni izlazni konektor, ali ima i digitalni pin te ugrađeni potencijometar i LM393 komparator napona. Potebno mu je vrijeme zagrijavanja kako bi ispravno radio. MQ senzori koriste se za detekciju otrovnih, bezopasnih i zapaljivih plinova. Senzor detektira NH₃, benzene, alkohol, dim, CO₂.

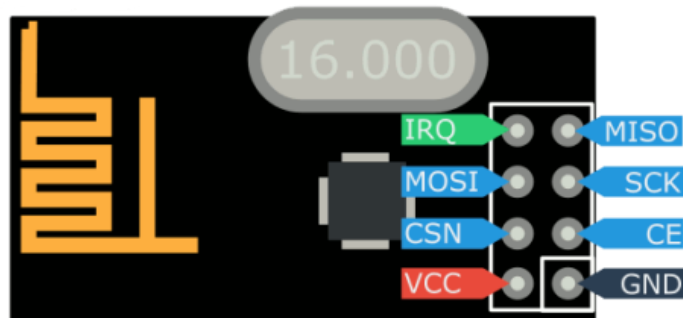


Slika 3.6. MQ135 senzor

3.1.7. NRF24L01+ 2.4Ghz radio modul

nRF24L01+ je novija verzija 2.4GHz radio modula, za razliku od starijeg nRF24L01 modula, podržava i 250 Kbps operacije nižih brzina. Iako modul nije ni najljepši ni najbolji ikad, pri vrlo maloj potrošnji, i niskoj cijeni daje izvrstan doomet koji iznosi više od 100 m na otvorenom području s ugrađenom antenom. Moguće ga je i povećati boljom, vanjskom antenom.

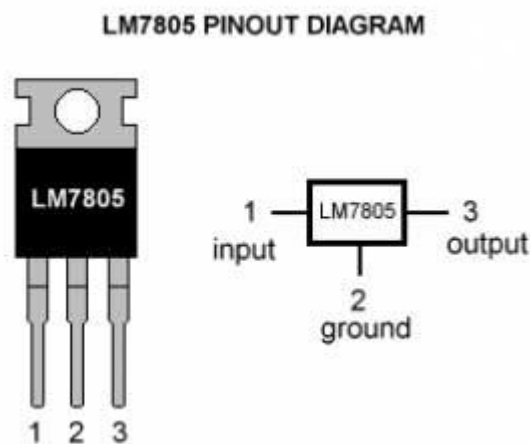
Ovaj modul nam je također pogodan jer je za njega izrađena biblioteka koja dizajnira mrežu u kojoj je jedan član baza a ostali nižeg ranga hijerarhije, kao u zadatku rada. Jedan NRF24L01 modul može aktivno slušati do 6 drugih. Tehničke karakteristike: **P 3.3.**



Slika 3.7. Konektori nRF24L01 modula

3.1.8. 7805 5V REGULATOR NAPONA THT

7805 vrlo se često koristi za 5V linearni regulator kada je to potrebno, dolazi u verziji montiranja kroz rupe (THT), ali i u drugim verzijama. Ti regulatori mogu pružiti lokalnu regulaciju, uklanjajući probleme distribucije povezane s regulacijom u jednoj točki. Sadrži ugrađeno unutarnje ograničenje struje, toplinsko isključivanje i zaštitu sigurnog područja, čineći ga u teoriji neuništivim. Ako se osigura odgovarajuće previsoke topline, može isporučiti preko 1A izlazne struje. Iako su dizajnirani prvenstveno kao regulatori s fiksnim naponom, ovi uređaji mogu se koristiti s vanjskim komponentama za dobivanje podesivih napona i struja.



Slika 3.8. 7805 regulator napona

3.1.9. Arduino IDE

Integrirano razvojno okruženje Arduino (IDE) je multiplatformna aplikacija (za Windows, macOS, Linux.) Koristi se za pisanje i prijenos programa na ploče kompatibilne s Arduinoom, ali također i drugih razvojnih ploča.

Arduino IDE podržava jezike C i C ++ koristeći posebna pravila strukturiranja koda. Korisnički napisan kod zahtijeva samo dvije osnovne funkcije, prvu za pokretanje skice i glavne programske petlje, koje se zatim prevode i povezuju u izvršni program, također uključeno u IDE okruženje. Arduino IDE napisani program učitava na ploču Arduino pomoću programa za učitavanje i same ploče.

3.1.10. Eagle

EAGLE je aplikacija za automatizaciju elektroničkog dizajna (EDA) sa shematskim prikazom, PCB tiskanim pločicama, značajkama automatskog usmjerivača i računalno potpomognute proizvodnje (CAM). **EAGLE** je skraćenica od Easily Applicable Graphical Layout Editor, a razvio ga je CadSoft. Tvrtku je 2016. godine kupio Autodesk Inc.

U radu je korišten u svrhu dizajniranja i izrade PCB tiskane pločice i za izradu shematskih prikaza.

3.2. Sklopovska realizacija

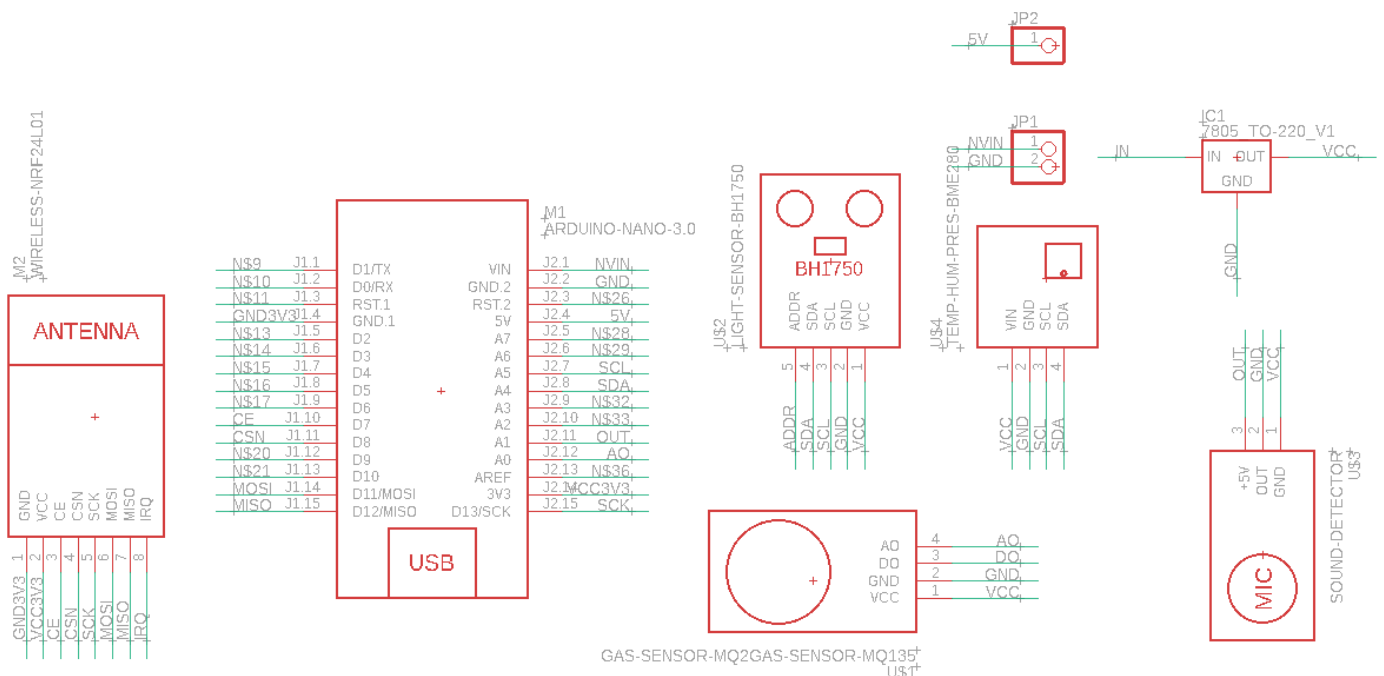
Realizacija sklopa svakog senzorskog čvora i centralne jedinice ovisi o brojnim faktorima. Prvotno je potrebno **odabrati senzore**. Na prijedlog mentora te iz osobnog iskustva i istraživanja odabrani su senzori koji mogu izmjeriti željene parametre. Presudni faktori za odabir određenih senzora bili su preciznost, dostupnost, cijena i kompaktnost. Zatim **odabrati prikladnu Arduino pločicu**, što ponajprije ovisi o potrebnom broju ulaznih i izlaznih pinova te kompaktnosti iste. Analizom tehničkih podataka svakog od senzora i modula za komunikaciju dobiva se broj potrebnih pinova kako bi se svi senzori uspjeli spojiti na jednu Arduino pločicu. Broj potrebnih analognih pinova iznosi 6, s time da senzor BME280 i BH1750 dijele dva zajednička konektora (isključivo SCL i SDA konektori na ploči), jer oba koriste I2C protokol za komunikaciju, Arduino pločice većinom imaju po jedan SCL i jedan SDA konektor, no svaki od senzora ima vlastitu jedinstvenu i različitu adresu što omogućuje da ih spojimo na iste te neće doći do miješanja podataka. Digitalni konektori su potrebni za NRF24L01 modul, koji koristi njih 5 (to su CSN, CE, SCK, MOSI i MISO.) Dobivši te podatke odabran je Arduino Nano kao ploča koja zadovoljava te potrebe i određene su veze sa sensorima. Točan prikaz veza u prilogu **P 3.4**.

Slijedeće, **odrediti i dizajnirati potrebna napajanja** za centralnu jedinicu te senzorski čvor. Centralna jedinica dobivat će napajanje pomoću AC-DC strujnog adaptera. Korišteni adapter daje izlazni napon od 5V i struju 2A što je prikladno za Raspberry Pi i Arduino Nano. Analizom tehničkih podataka senzora utvrđeno je da svi senzori zahtijevaju napon od 5V. Potrošnja struje prilikom mjerenja provjerena je u tehničkim podacima i dodatno izmjerena pomoću multimetra za svaki senzor zasebno tako što je multimetar podešen u način rada ampermetra i spojen u seriji sa sensorom. Očitano je da svi senzori imaju vrlo malu potrošnju, u rangju manjem od 10mA, osim senzora MQ135 za koji je izmjereno da u fazi zagrijavanja troši do 180mA. Za napajanje čvorova

korištene su 4 baterije tipa AA koje daju napon od oko 6V, te 7805 regulator napona na čiji su ulaz baterije spojene, a na izlaz, koji daje pripadajućih 5V su spojeni svi senzori te preporučeni kondenzator. Razlika u naponu nije velika što je poželjno jer nema puno gubitaka prilikom reduciranja. Na regulator napona može se dodati komad metala dizajniran za raspodjelu topline. Arduino Nano također radi na naponu od 5V, što je ostvareno pomoću 9V baterije spojene na VIN ulaz na pločici koji za koji je preporučeno da napon bude između 7V i 12V koji se zatim reducira na potrebnih 5V. Modul za komunikaciju zahtjeva 3.3V te će on dobiti napajanje iz Arduino Nano-a, konektor 3V3 koji daje 3.3V. Zbog kompaktnosti i urednosti korišteno je kućište za 4 AA baterije i konektor za 9V bateriju.

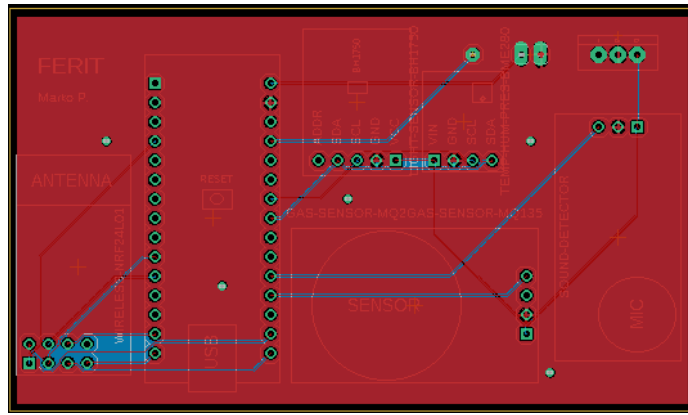
Uzeći sve u obzir izrađen je prvi prototip na praznoj tiskanoj pločici. Arduino Nano, svi senzori, modul za komunikaciju i napajanja su povezani i obavljeno je testiranje.

Nakon toga, po uzoru na izrađeni prototip **dizajniran je i izrađen PCB** senzorskog čvora za ovaj projekt. Korišten je navedeni program Eagle. Prvi korak je stvoriti novi projekt, zatim u shematski prikaz dodane su komponente koje će se nalaziti na tiskanoj ploči, to je dalo otisak potrebnih rupa za bušenje i olakšalo izradu rasporeda. To uključuje Arduino Nano, sve senzore i modul za komunikaciju. Stvorene su i dodatne rupe kako bi se mogli spojiti regulator napona i konektori za baterije.

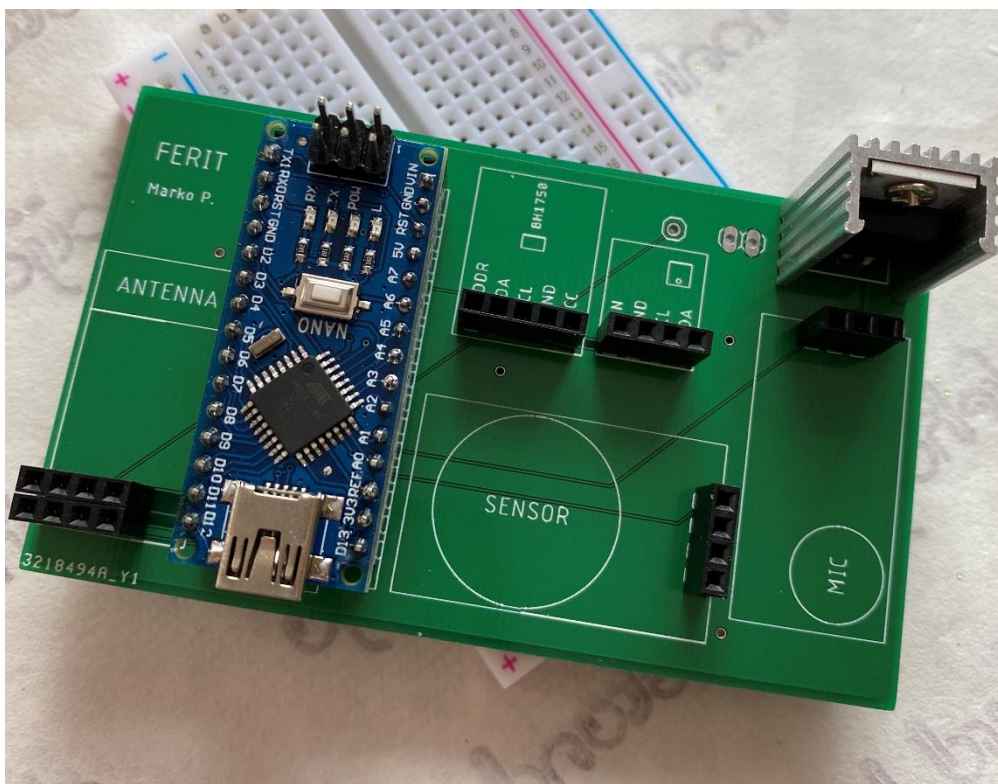


Slika 3.9. shematski prikaz

U shematskom prikazu ostvarene su veze između željenih pinova tako što su nazvane istim imenom (slika 3.9. zelene linije) te Eagle automatski ponudi da ih poveže, što olakšava posao kasnije. Primjer, sva napajanja su dovedena na izvor od 5V tako što su nazvana istim imenom „VCC“ i prihvaćena je ponuđenu opcija da se povežu. S gotovom shemom prebacuje se na prikaz PCB-a, ondje su svi elementi koji su prvotno dodani u shematski prikaz s točnim dimenzijama i vezama. Komponente se povlače na prostor za PCB te raspoređuju po želji, potrebno je paziti na zbijenost senzora i orijentaciju, posebice temperature i svjetlosti. Nakon raspoređivanja pokrenut je auto-router, pomagalo programa Eagle, koji određuje najpovoljnije rute i položaje za ostvarivanje veza koristeći gornji i donji sloj PCB-a. Samostalno radimo korekcije ako je to potrebno. Kako bi projekt mogao biti poslan na izradu stvorene su gerber datoteke i poslane na izradu web servisu JLCPCB. Na izrađenom PCB-u testirane su veze te su dodani konektori i sve komponente te je rad senzorske jedinice testiran pisanim algoritmom.



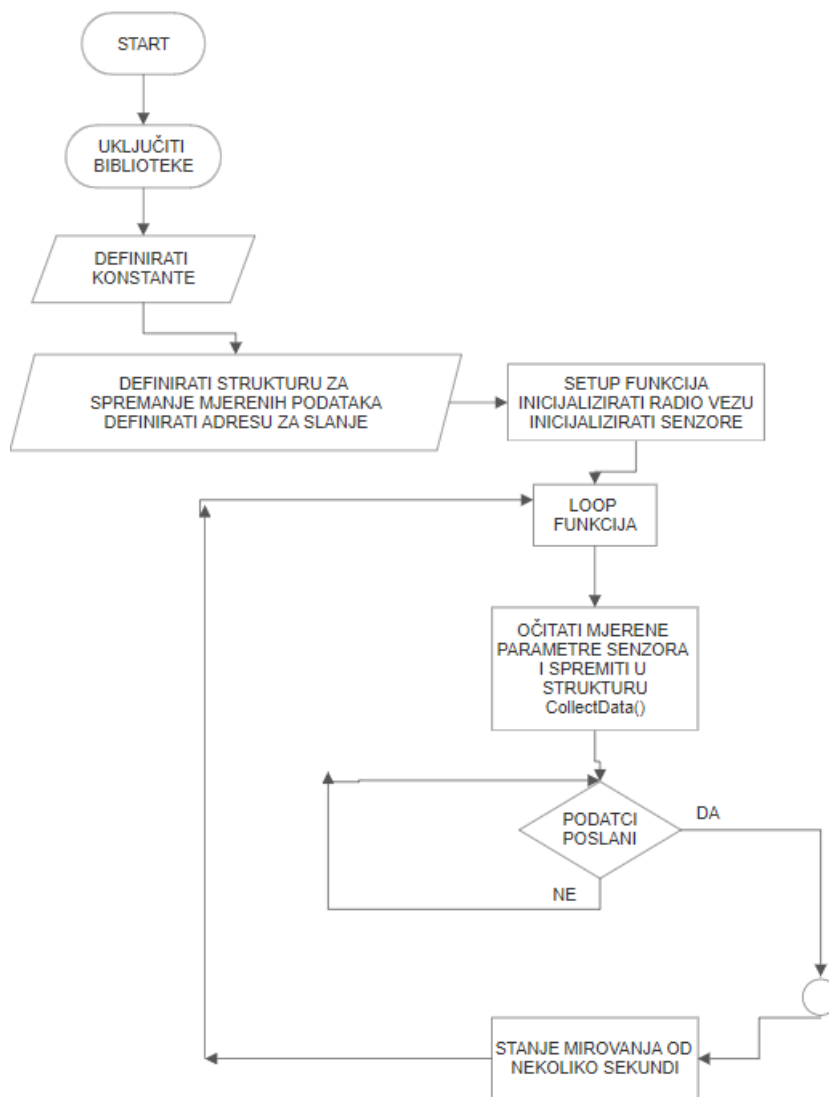
Slika 3.10. Dizajnirani PCB



Slika 3.11. Izradeni PCB

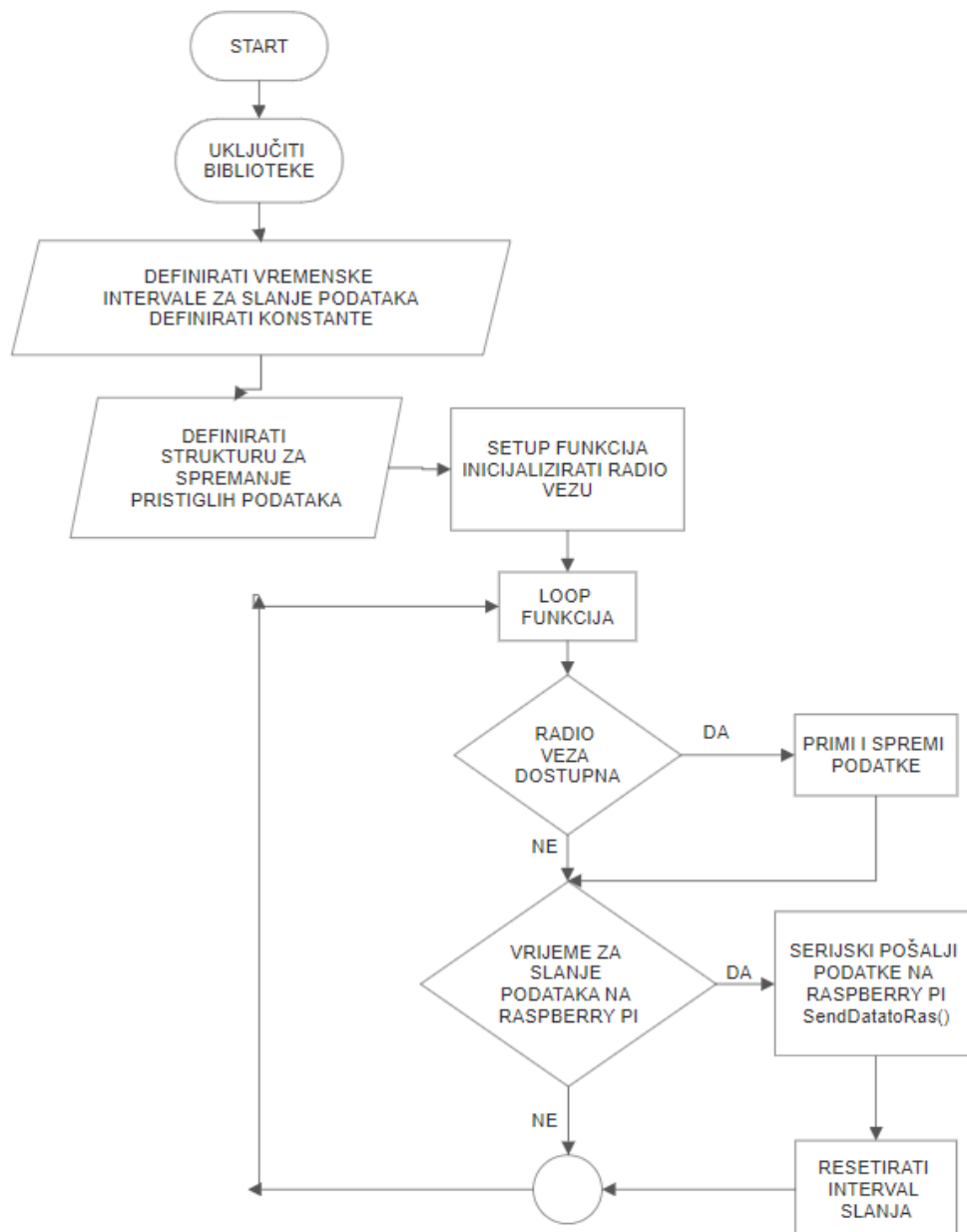
3.3. Algoritamska realizacija

Nakon dovršene sklopovske realizacije svaka senzorska jedinica i centralna jedinica su zasebno programirane. Pisani algoritmi izvršavaju zadatke uspješno kao što je zamišljeno i potrebno za ispravan rad susatva. **Algoritmi za senzorske** jedinice su vrlo slični, jedina promjena je ID senzorske jedinice i put za slanje podataka bežičnim modulom prema centralnoj jedinici. Kod algoritma jedne od senzorskih jedinica s komentarima dan je u prilogu **P 3.5**. Okvirna procedura i tijek algoritma senzorske jedinice prikazani su blok dijagramom na slici ispod.

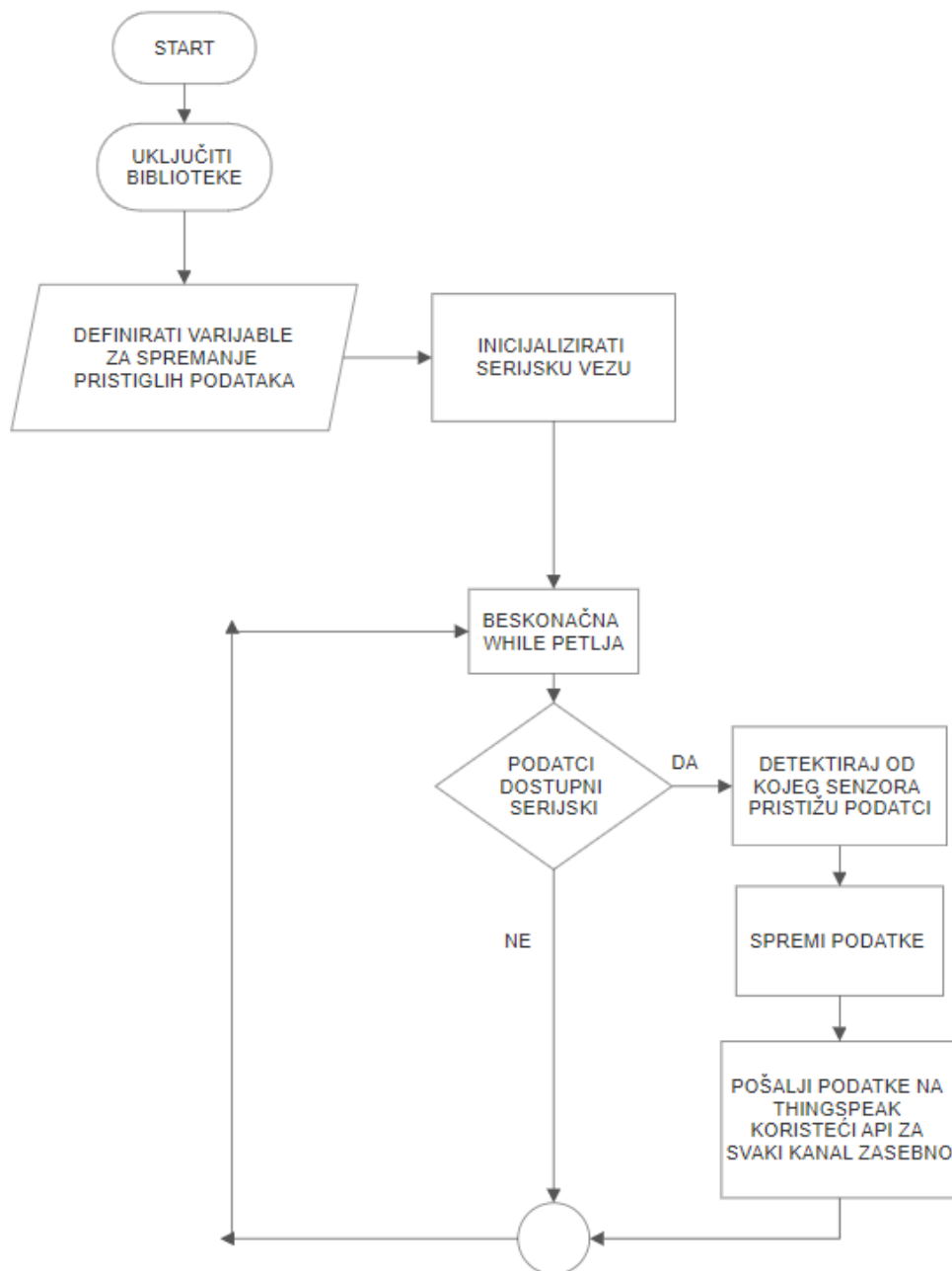


Slika 3.12. Blok dijagram toka algoritma senzorske jedinice

Algoritam centralne jedinice ovisi o algoritmu senzorskih jedinica. Kod algoritma centralne jedinice podijeljen je u dvije komponente kao i sama struktura sklopa. Prvi dio na koji pristižu poslani podatci putem bežičnog prijemnog modula je Arduino Nano koji te podatke zatim šalje serijski određenim redoslijedom na Raspberry Pi. Tada se podatci identificiraju, spremaju točnim redoslijedom kako pristižu u polja podataka određenog tipa te kao zadnji korak, podižu na IoT ThingSpeak platformu koristeći jedinstveni API kod za kanal svakog senzorskog čvora. Kod algoritma centralne jedinice Arduino Nano dijela dan je u prilogu **P 3.6.**, a kod Python skripte pisane za Raspberry Pi u prilogu **P 3.7.**



Slika 3.13. Blok dijagram toka algoritma Arduino Nano centralne jedinice



Slika 3.14. Blok dijagram toka algoritma Raspberry Pi centralne jedinice

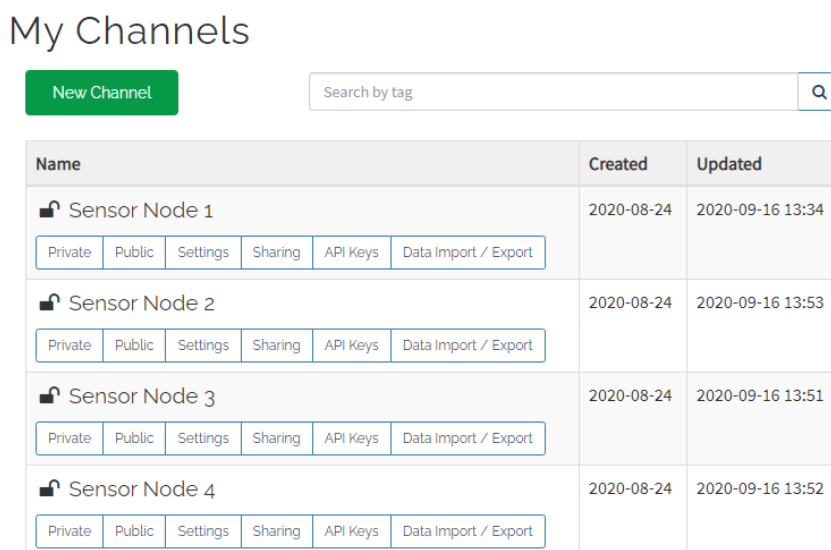
3.4. Sučelje čovjek-stroj i komunikacijska infrastruktura

Web IoT platforma koja će služiti kao sučelje između korisnika i mjerenih podataka je ThingSpeak. Navedena platforma je besplatna do određene mjere i s određenim ograničenjima. ThingSpeak omogućuje agregiranje, vizualizaciju i analizu tokova podataka uživo u oblaku. Uz ThingSpeak podaci se pohranjuju u kanale. Svaki kanal pohranjuje do 8 polja podataka. Jedna od glavnih značajki je da daljinski vizualiziramo podatke senzora u stvarnom vremenu. ThingSpeak automatski ucrtava podatke koji su poslani, tako da je moguće daljinski nadgledati uređaje ili opremu s bilo kojeg mjesta. Podaci se mogu pregledati na bilo kojem web pregledniku ili mobilnom uređaju.

Pomoću mehanizma MATLAB ugrađenog u ThingSpeak moguće je izvoditi kalibracije, razvijati analitiku i transformirati IoT podatke. Za izradu prilagođenih grafikona može se koristiti i mehanizam MATLAB ugrađen u ThingSpeak. S komercijalnom licencom moguće je pokretati izračune MATLAB-a koji traju do 60 sekundi. Komercijalna licenca također omogućuje upotrebu MATLAB-a za strojno učenje, obradu signala, identifikaciju sustava i još mnogo toga.

Pomoću aplikacije Time Control možete zakazati da se izračuni izvršavaju jednom dnevno, jednom u satu ili najbrže jednom u 5 minuta. Aplikacija React koristi se za praćenje stanja. Možete nadzirati podatke koji dolaze s uređaja i postaviti upozorenje kada podaci ukazuju na to da bi nešto trebalo pažnju. Na primjer, moguće je konfigurirati ThingSpeak za slanje e-pošte kada vlaga na tlu biljke premaši određenu vrijednost.

Za svaku senzorsku jedinicu stvoren je po jedan kanal i 7 polja podataka, po jedan za svaki mjereni parametar.



The screenshot shows the 'My Channels' interface on the ThingSpeak platform. At the top, there is a 'New Channel' button and a search bar labeled 'Search by tag'. Below this is a table with four rows, each representing a sensor node. Each row includes a name, a 'Created' date, and an 'Updated' date and time. Below the name in each row are several buttons: 'Private', 'Public', 'Settings', 'Sharing', 'API Keys', and 'Data Import / Export'.

| Name | Created | Updated |
|--|------------|------------------|
| Sensor Node 1 Private Public Settings Sharing API Keys Data Import / Export | 2020-08-24 | 2020-09-16 13:34 |
| Sensor Node 2 Private Public Settings Sharing API Keys Data Import / Export | 2020-08-24 | 2020-09-16 13:53 |
| Sensor Node 3 Private Public Settings Sharing API Keys Data Import / Export | 2020-08-24 | 2020-09-16 13:51 |
| Sensor Node 4 Private Public Settings Sharing API Keys Data Import / Export | 2020-08-24 | 2020-09-16 13:52 |

Slika 3.15. Prikaz stvorenih kanala na ThingSpeak platformi

Svaki kanal pa tako i polje podataka zasebno moguće je dodatno postaviti i urediti prema željenim postavkama. Odabrana platforma automatski će vizualizirati primljene podatke i prikazati ih u odgovarajućem polju. Podaci tada postaju lako čitljivi korisniku u gotovo stvarnom vremenu kada su očitani i primljeni. Dodatno, svaki kanal zasebno bilježi broj unosa. Svaki graf za pojedini mjereni parametar također je moguće urediti po željenim opcijama, od boje i vrste grafa, do vremenskog perioda koji se prikazuje na grafu.

Channel Stats

Created: 28 days ago
Last entry: 5 days ago
Entries: 316



Slika 3.16. Prikaz jednog od kanala i šest polja mjerenih parametara

4. TESTIRANJE I REZULTATI

4.1. Metode i načini testiranja

Kako bi izrađeni sustav mjernih jedinica i centralne jedinice bio što potpuniji i detaljnije shvaćen i razumljiv kako od nas tako i od krajnjeg korisnika obavljena su testiranja određenih parametara i ponašanja rada sustava. Testiranje je podijeljeno u zamišljene dvije podskupine, testiranje samog fizičkog sklopa i algoritma. Testiranje je prvo obavljeno za sklopovski dio sustava, što je obuhvatilo provjeru veza i ispravnosti ruta na izrađenoj PCB pločici, potrošnju struje senzora te testiranje baterijskog napajanja. Kako bi se ova testiranja obavila što bolje korišten je digitalni multimeter kao alat za očitavanje željenih vrijednosti i određeni izračuni.

Nakon dobivene povratne informacije te korekcije nekih od parametara ako je to bilo potrebno testira se algoritamski dio sustava. Tim dijelom obuhvaćeno je testiranje točnosti mjerenih parametara i ponašanja svakog od senzora u različitim pruženim situacijama, što će biti prikazano u usporedbi sa stvarnim pouzdanim podacima, iduće se testira ostvarivanje bežične komunikacije između senzorske i centralne jedinice te pouzdanost i domet iste, prikazujući vrijeme kašnjenja ako postoji te brzinu slanja podataka, u tome nam pomaže Arduino Serial monitor ugrađen u okruženje za pisanje koda koji nam daje točno vrijeme određenog događaja. Nakon toga obavljeno je i završno testiranje postavljanja i pokretanja kompletnog sustava.

4.2. Rezultati testiranja

Testiranje fizičkih veza i ruta na PCB tiskanoj pločici obavljeno je tako što se digitalni multimeter postavio u način za ispitivanja kontinuiteta koji će kada se testni vodiči iz multimetra dotaknu, postaviti na žicu ili vezu koja povezuje dvije točke na PCB-u aktivirati zvučni signal i lampicu. Taj događaj daje nam povratnu informaciju da su dotaknute točke ispravno povezane. Testirali smo povezanost svakog senzora sa pripadajućim Arduino Nano pinom te dovod do napajanja. Utvrđena je potpuna ispravnost svih 5 PCB tiskanih pločica te daljnje radnje nisu bile potrebne.

Nakon toga, digitalni multimeter prebacujemo u način rada ampermetra i spajamo ga u seriji sa svakim sensorom zasebno i modulom za komunikaciju kako bi dobili informaciju o potrošnji. U tablici su prikazane izmjerene potrošnje, provjereno je u tehničkim podacima svakog senzora te utvrđeno da su izmjerene vrijednosti ispravne.

| Komponenta | Potrošnja prilikom rada | Potrošnja u mirovanju |
|----------------|-------------------------|-----------------------|
| NRF24L01 | 13,5 mA | 900 nA |
| MIKROFON LM393 | 5 mA | - |
| BH1750 modul | 190 μ A | 1 μ A |
| BME280 modul | 3.6 μ A | 0.2 μ A |
| MQ135 modul | <160 mA | - |

Tablica 3.1. izmjerena potrošnja komponenti

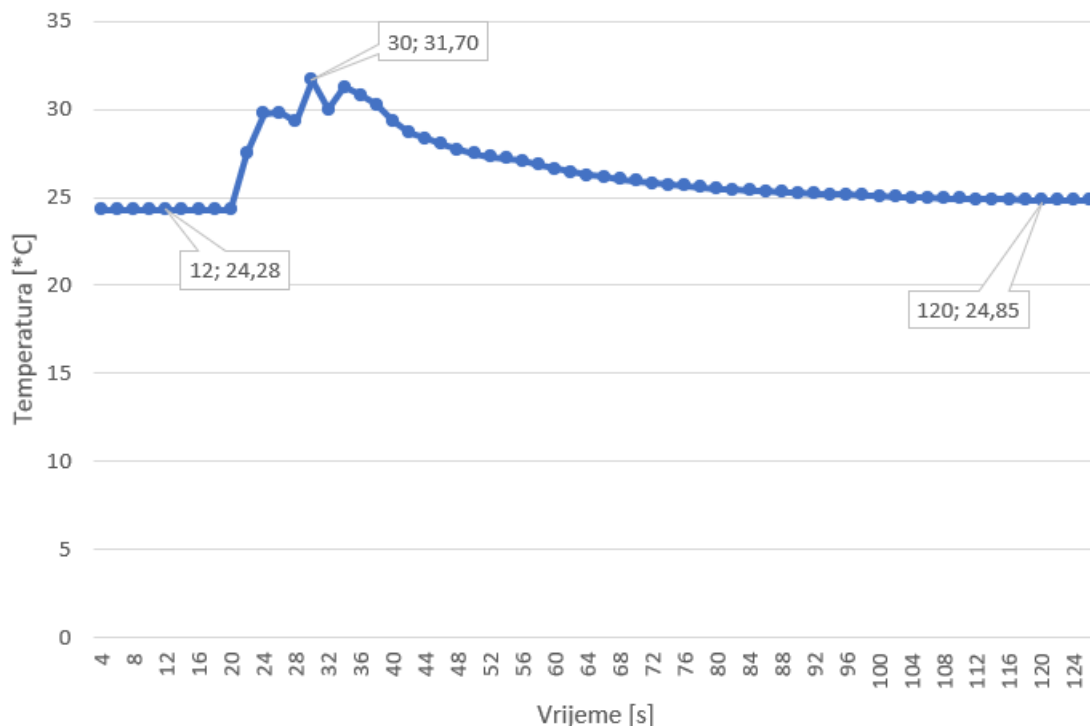
Pomoću potrošnje možemo izračunati okvirno trajanje baterijskog napajanja za senzorsku jedinicu. Prilikom računanja ne uzimamo u obzir NRF24L01 modul jer on dobiva napajanje iz Arduino Nano pločice te također MQ135 jer ima visoku potrošnju prilikom zagrijavanja. Baterijsko napajanje čine 4 alkaline baterije modela AA, istraživanjem je utvrđeno da AA alkaline baterije imaju kapacitet od 1700-2850 mAh. Kako je raspon prilično širok uzet ćemo srednju vrijednost.

$$= \frac{(2850mAh - 1750mAh) : 2 * 4}{5mA + 0,19mA + 0,036mA}$$

Kada podijelimo kapacitet 4 AA baterije sa aktivnom potrošnjom senzora, dobijemo brojku od 420, 97 h aktivnog mjerenja. Ta vrijednost provjerit će se tek stvarnom upotrebom jer se baterije ne ponašaju jednako u svim uvjetima potrošnje, neke će trajati i znatno duže pri niskoj potrošnji dok neke mogu dati visoku struju za kraće vrijeme, a također i izmjerena potrošnja senzora nije u potpunosti pouzdana.

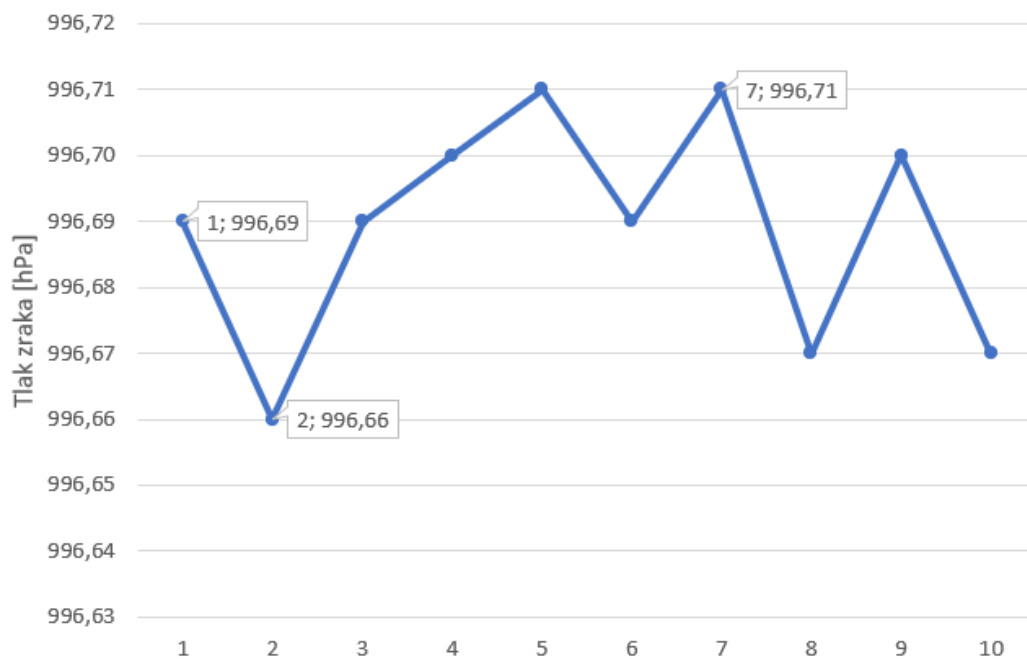
Nadalje, mjerili smo točnost i ponašanje svakog senzora u danim uvjetima tako što su spojeni žicama za prototipiranje na Arduino Nano koji je programiran da očita mjerenu vrijednost i ispiše na Serial monitor svake dvije sekunde.

Očitavanje vrijednosti **temperature** BME280 modula uspoređeno je sa kućnim digitalnim termostatom, oboje su postavljeni u istu sobu i ostavljeni neko vrijeme da se prilagode. Za kućni digitalni senzor očitana je vrijednost od 24,2 °C, što je uzeto za referentnu vrijednost. Vrijednosti izmjerene BME280 modulom prikazane su na slici ispod. Vidimo kako senzor mjeri 24,28 °C u početku, što nam daje informaciju da je izmjerena vrijednost točna i u razmjerima pogreške koje je naveo proizvođač od ± 0.5 °C. Nakon 10 izmjerenih vrijednosti dovodimo izvor topline na senzor na 10 sekundi te temperatura raste do izmjerenih 31,70 °C, nakon toga odmiče se izvor topline te je senzoru nakon događaja nagle porasti temperature potrebno neko vrijeme da ponovno mjeri temperaturu prostorije što možemo vidjeti i na grafu na slici, otprilike 40-50 sekundi postupnog smanjenja temperature. Iz toga zaključujemo da pri ekstremnim promjenama senzoru je potrebno neko vrijeme da se vrati u prvobitno stanje.



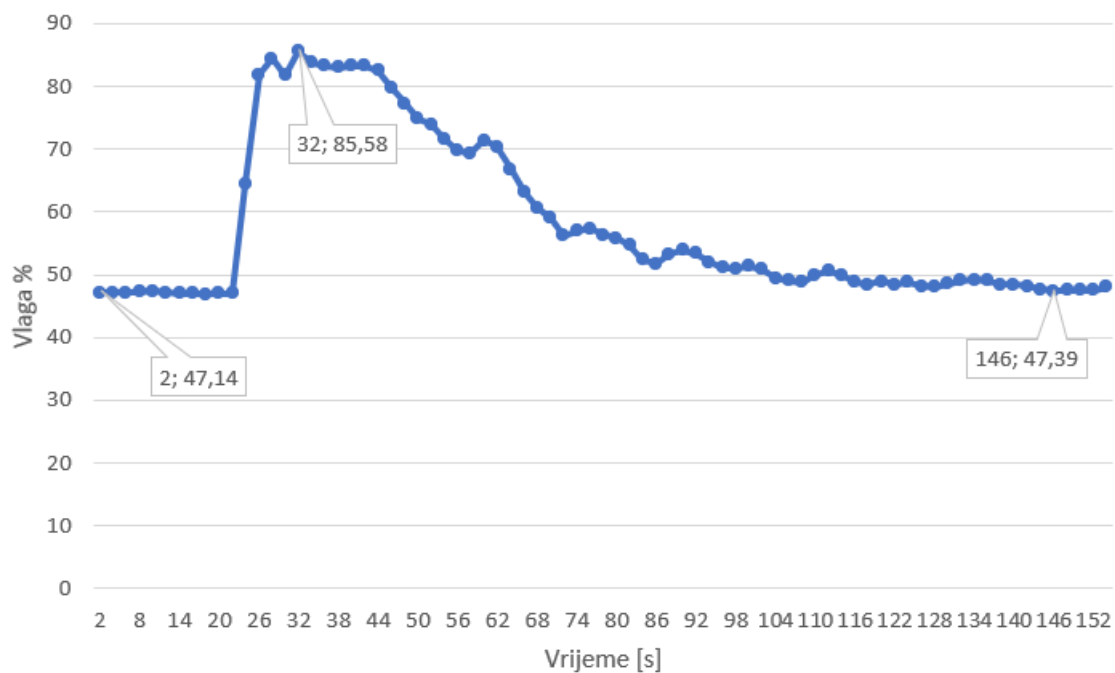
Slika 4.1. izmjerene vrijednosti temperature

Pomoću BME280 modula također mjerimo **tlak zraka** u hPa, kako bi testirali točnost poslužiti ćemo se trenutnim podacima Državnog hidrometeorološkog zavoda, sa web stranica možemo vidjeti da vrijednost tlaka zraka na našem području u trenutku testiranja iznosi 1013,9 hPa. Izmjerene vrijednosti senzora prikazane su na slici ispod. Nije potrebno obaviti velik broj uzoraka jer su očitane vrijednosti konzistentne i bez velikih skokova. Vidimo da postoji pogreška u očitavanju senzora za oko 20 hPa, u tehničkim podacima navedena je preciznost od ± 1 hPa tako da je potrebno obaviti kalibriranje ili promjene u biblioteci senzora kako bi dobili točnije vrijednosti.



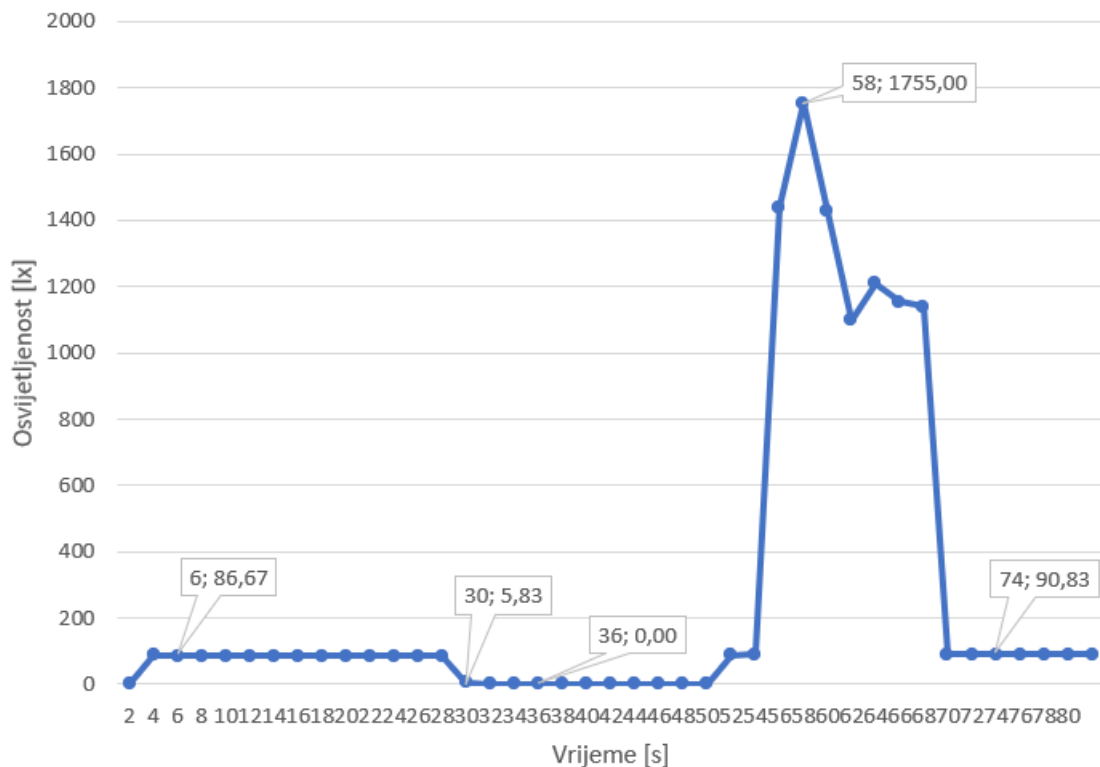
Slika 4.2. izmjerene vrijednosti tlaka

Relativnu vlažnost također mjerimo BME280 modulom prikazano u postotku (0-100%). Senzor je postavljen u zatvorenu prostoriju te ostavljen neko vrijeme da se prilagodi. Nakon toga očitane su vrijednosti uz pomoć Arduina Nano, zabilježene, te je u senzor puhanu tokom 10 sekundi, u tom razdoblju dolazi do znatnog povećanja vlažnosti što vidimo na grafu ispod, nakon toga izlaganja povećanoj vlazi senzoru je potrebno neko vrijeme da se vrati u stanje u kojemu mjeri vlažnost prostorije, oko 40-50 sekundi.



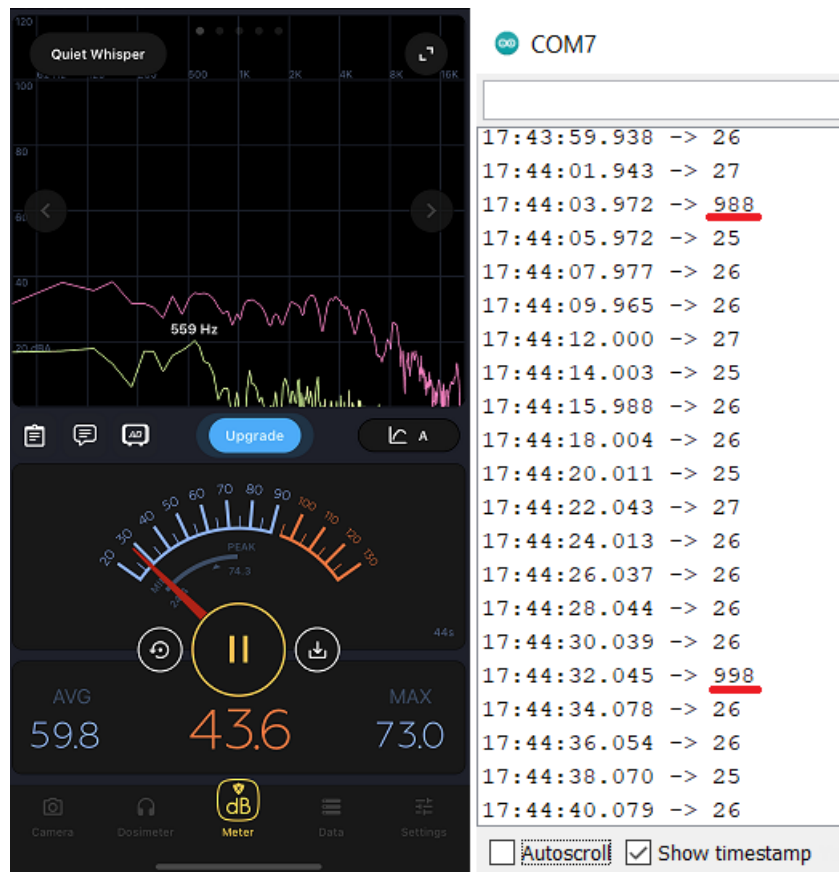
Slika 4.3. izmjerene vrijednosti vlage prilikom testiranja

Intenzitet osvjetljenja mjerimo BH1750 modulom, prikazanu mjernom jedinicom lx (luks.) Određen je osvjetljenjem plohe kojoj je na četvorni metar jednoliko raspoređen svjetlosni tok od jednoga lumena (lm.) Prilikom testiranja senzor je postavljen u zatvorenu prostoriju bez vanjskog osvjetljenja, smješten pod stolnu lampu. U tom stanju izmjereno je 10 vrijednosti, koje iznose konstanto 86,67 lx. Nakon toga lampa je ugašena, te su vrijednosti ponovno izmjerene i iznosile su 5,83 lx. Nakon toga senzor smo prekrili kako bi uklonili sve izvore svjetlosti na površinu i očitali izmjerene vrijednosti od 0 lx. Ponovno palimo lampu i dodajemo bljeskalicu od mobilnog uređaja prema senzoru, te očitavamo vrijednosti do 1755 lx. Nakon toga gasimo bljeskalicu te se senzor vraća u prvobitno stanje pod sobnom lampom, te očitane vrijednosti iznose 90,83 lx. Primjećuje se kako se za razliku od prijašnjih mjerenih senzora, ovaj senzor puno brže prilagođava na promjene i ekstremne situacije. Prikaz postupka na grafu ispod. Očitane vrijednosti su pod sobnom lampom, dok pod sunčevom svjetlosti ili jačim izvorom vrijednosti mogu iznositi i >20000 lx.



Slika 4.4. testiranje mjerenja intenziteta osvjetljenja

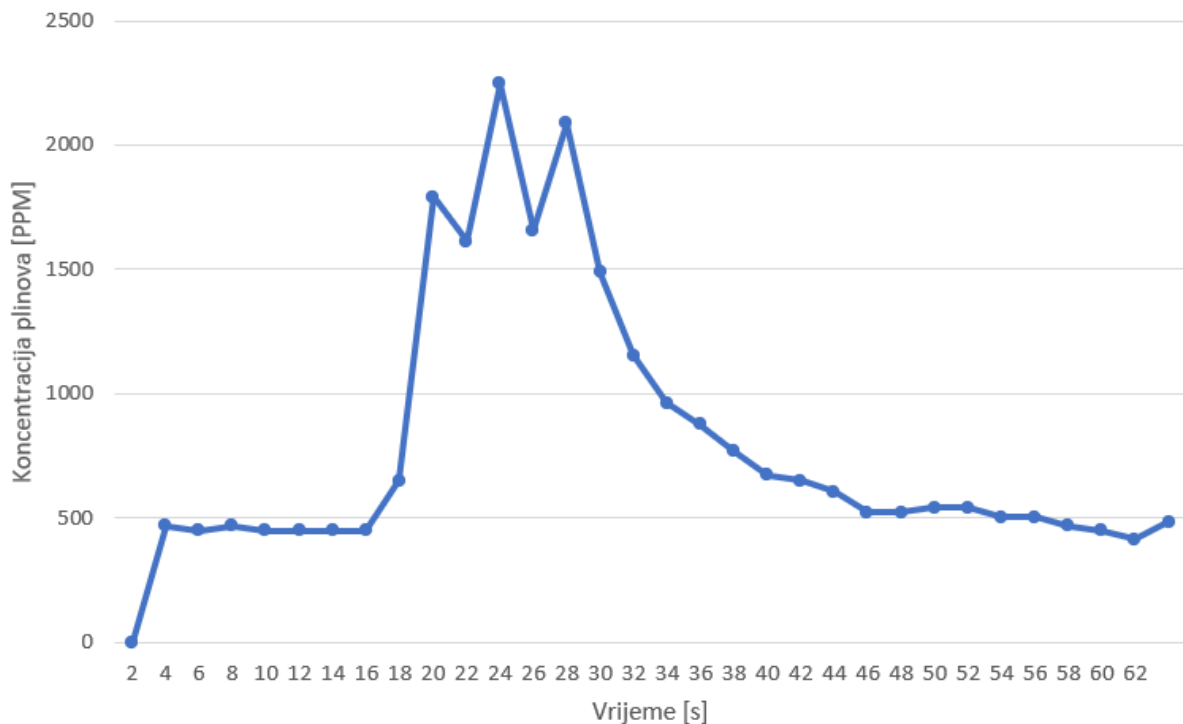
Za mjerenje **intenziteta buke** odabrali smo mikrofonski modul s LM393. Prilikom testiranja dolazimo do zapažanja da odabrani modul ne zadovoljava željene potrebe. Zamisljeno je detektirati razinu buke u čovjekovom spektru sluha, za što bi ovaj mikrofonski modul i analogni izlaz trebali biti prikladni. No ponašanje senzora je jako grubo i naglo, iako ima analogni izlaz gotovo kao da može očitati samo dvije vrijednosti, jednu u mirnom stanju, od 24-29 dB, te drugu kada se dogodi pojava buke, a tada sensor očitava iznimno velike i netočne vrijednosti od oko 900 dB. Kako bi imali neku referentnu vrijednost za usporediti koristi se mobilna aplikacija za mjerenje buke. Postupak je prikazan na slici ispod, vidimo kako aplikacija prilikom povećavanja buke postupno mjeri očekivano umjereno povećanje, do 70 dB. Ovaj modul ne preporučujem za ovu upotrebu te će isti biti zamijenjen, ovaj modul mogao bi pronaći svoju primjenu u drugim situacijama gdje je potrebno reagirati na naglu promjenu buke.



Slika 4.5. usporedba vrijednosti mikrofona mobilnog uređaja(lijevo) i korištenog senzora(desno)

Posljednji testirani senzor je MQ135, senzor **koncentracije plinova**. Senzor reagira na koncentraciju određenih plinova u zraku. Detektira plinove poput dušika, amonijaka, kisika, alkohola, aromatičnih spojeva, sulfida i dima. MQ-135 senzor za plin može biti izvedba za otkrivanje dima, benzena, pare i drugih štetnih plinova no u našem sustavu prikazivat će koncentraciju svih plinova izraženu u PPM (eng.: parts per million). Jedan ppm predstavlja jedan dio na milijun. Kako bi senzor počeo s radom potrebno je napraviti neke izmjene u biblioteci, to je otkriveno tako što su očitane vrijednosti bile vrlo pogrešne. Prvo što je otkriveno da je na modulu korišten otpornik od 1kOhma umjesto 10kOhma kako je navedeno u varijabli u biblioteci „MQ135.h.“ Nakon toga senzor je potrebno ostaviti priključenog na napajanje 48h kako bi se zagrijavanjem kalibrirao te rezultati postali stabilniji. Nakon toga priključiti Arduino mikrokontroler, izmjeriti i očitati vrijednost Rload konstante koja se izmjeni u biblioteci, dobiveno je 92,37. Nakon tih promjena u biblioteci senzor počinje davati točnije i stabilnije rezultate. Kako bi usporedili točnost za referentnu vrijednost koju pronalazimo pretraživanjem interneta uzet ćemo trenutnu koncentraciju CO2 u atmosferi koja iznosi 412.55.

Na grafu ispod prikazane su izmjerene vrijednosti, vidljivo je da senzor u zatvorenom prostoru očitava nešto manje od 500 ppm. Nakon toga kako bi testirali reagiranje senzora na promjene puhnemo nekoliko sekundi u područje mjerenja. Dolazi do očekivanog povećanja vrijednosti te se mjerene vrijednosti nakon nekog vremena, od oko 20 sekundi, vraćaju u prvobitno stanje. Modul će se koristiti za opću kvalitetu zraka u prostoru postavljanja što će dati iskoristive rezultate.



Slika 4.6. izmjerene vrijednosti koncentracije plinova u zraku

Po završetku testiranja senzora, montiraju se na izrađeni PCB te se testira **bežična komunikacija** između modula i centralne jedinice. Prvi testirani parametar je **domet** na kojemu se može ostvariti pouzdana bežična komunikacija tako što će jedna senzorska jedinica slati mjerene podatke, spremljene u strukturu veličine 28 bajta. Maksimalna veličina koju ovaj modul može poslati u jednom paketu iznosi 32 bajta. Centralna jedinica ispisati će svakih 5 sekundi primljene podatke te vrijeme kada su primljeni. Za uspješno primanje podataka uzet će se samo kompletni pristigli podatci u željenom vremenu. Test je izveden na otvorenom prostoru bez prepreka između komunikacijskih modula, tako će biti korišteni i u stvarnoj primjeni. Za rezultat dometa proizvođač navodi ~800 m izravne zračne linije bez prepreka, no to ovisi i o postavljenoj brzini prijenosa, gdje je 250 Kbps najmanja moguća kada će senzor biti najosjetljiviji, i 2 Mbps kada će brzina prijenosa

biti puno veća no senzor neće biti toliko osjetljiv, odabrana je brzina 2 Mbps jer nema potrebe za velikom udaljenosti, i pri programiranju algoritma je uočeno da ovi moduli rade bolje i na manjim udaljenostima pri tim brzinama. Pri odabranoj brzini paketi veličine 28 bajta bi trebali biti poslani u cijelosti bez ikakvih problema. Prilikom testiranja postepeno se povećavala udaljenost između dva modula te očitao nekoliko pristiglih paketa da se utvrdi da su uspješno primljeni bez kašnjenja. Kada je udaljenost bila >100m i podatci su uspješno pristizali, ponestalo je prostora gdje bi moduli i dalje bili jedno drugom u vidnom polju te smo ga premjestili iza prepreke. Rezultat toga bio je prestanak uspješnog primanja podataka. Iz toga zaključuje se da moduli rade puno bolje dok su jedno drugome u vidnom polju bez prepreka, i da na manjim udaljenostima prepreke nisu značajne kao na većim. Treba također napomenuti da je test izveden na području bez značajnih signala koji bi ometali module koji rade na 2.4 Ghz kao i neki WIFI te kućni uređaji te im je kanal postavljen na 108, što je iznad većine spomenutih WIFI uređaja.

Kako bi testirali složenost **postavljanja sustava** sve jedinice su ugašene i nisu spojene na napajanja. Nalazimo se u situaciji krajnjeg korisnika koji će senzorske jedinice postaviti na željena mjesta, priključiti baterije kao izvor te pritisnuti tipku RESET na Arduino Nano kao dodatnu mjera sigurnog pokretanja. Centralnu jedinicu postavlja na razumnu udaljenost od senzorskih, te joj se pruža stabilna internetska veza žičanim putem i priključuje na napajanje što će rezultirati u pokretanju Rasperry Pi-a, pokretanja Python skripte za slanje mjerenih podataka na IoT ThingSpeak platformu i time uspješno završava postavljane cijelog sustava.

5. ZAKLJUČAK

Uspješno obavljeno cilj rada bio je objasniti, dizajnirati i izraditi sustav senzorskih jedinica koji mjerene odabrane parametre preko centralne jedinice i odbrane IoT tehnologije prikazuju korisniku. U radu je prikazano kako uz pomoć Arduino Nano mikrokontrolera upravljati i iskoristiti senzore te pomoću Raspberry Pi-a podignuti podatke na IoT web platformu. Na kraju rada se može vidjeti obavljeno testiranje, kako se sustav ponaša pri zadanim ciljevima te problemi koje je moguće susresti. Jedna od velikih pogodnosti sustava je mogućnost proširenja i napredovanja sustava, kako dodavanja senzorskih jedinica, tako i dodavanja novih senzora i aktuatora na već postojeće senzorske jedinice. Izrađeni sustav može pronaći primjenu u raznim kategorijama, od kućne do upotrebe u sustavima plasteničke ili stakleničke proizvodnje.

LITERATURA

[1] TEO

<https://www.slideserve.com/keira/environmental-monitoring-using-sensor-networks-texas-environmental-observatory> , pristup: 12.07.2020.

[2] MoteLab, IEEE

<https://ieeexplore.ieee.org/document/1440979> , pristup: 12.07.2020.

[3] Wikipedia, Vlažnost zraka

https://hr.wikipedia.org/wiki/Vla%C5%BEnost_zraka

[4] Wikipedia, Tlak

https://hr.wikipedia.org/wiki/Atmosferski_tlak

[5] E-radionica, vrste i karakteristike svjetla

<https://e-radionica.com/hr/blog/2015/08/19/vrste-i-karakteristike-svjetla/> , pristup: 13.07.2020.

[6] Wikipedia, Fotodioda

<https://hr.wikipedia.org/wiki/Fotodioda>, pristup: 18.09.2020.

[7] E-radionica, BME280

<https://e-radionica.com/hr/blog/2018/03/24/kkm-bme280/> , pristup: 14.7.2020.

PRILOZI

3.1. Arduino Nano

| | |
|-------------------------|--|
| Microcontroller | ATmega328 |
| Architecture | AVR |
| Operating Voltage | 5 V |
| Flash Memory | 32 KB of which 2 KB used by bootloader |
| SRAM | 2 KB |
| Clock Speed | 16 MHz |
| Analog IN Pins | 8 |
| EEPROM | 1 KB |
| DC Current per I/O Pins | 40 mA (I/O Pins) |
| Input Voltage | 7-12 V |
| Digital I/O Pins | 22 (6 of which are PWM) |
| PWM Output | 6 |
| Power Consumption | 19 mA |
| PCB Size | 18 x 45 mm |
| Weight | 7 g |
| Product Code | A000005 |

3.2. Raspberry Pi Model B+

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

3.3. NRF24L01+

Key Features

- Worldwide 2.4GHz ISM band operation
- 250kbps, 1Mbps and 2Mbps on air data rates
- Ultra low power operation
- 11.3mA TX at 0dBm output power
- 13.5mA RX at 2Mbps air data rate
- 900nA in power down
- 26µA in standby-I
- On chip voltage regulator
- 1.9 to 3.6V supply range
- Enhanced ShockBurst™
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™
- Drop-in compatibility with nRF24L01
- On-air compatible in 250kbps and 1Mbps with nRF2401A, nRF2402, nRF24E1 and nRF24E2
- Low cost BOM
- ±60ppm 16MHz crystal
- 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

Applications

- Wireless PC Peripherals
- Mouse, keyboards and remotes
- 3-in-1 desktop bundles
- Advanced Media center remote controls
- VoIP headsets
- Game controllers
- Sports watches and sensors
- RF remote controls for consumer electronics
- Home and commercial automation
- Ultra low power sensor networks
- Active RFID
- Asset tracking systems
- Toys

3.4. Veze između senzora i Arduino Nano pinova

| | |
|------------------|-------------|
| BH1750 | NANO |
| VCC | 5V |
| GND | GND |
| SCL | A5 |
| SDA | A4 |
| BME280 | |
| VCC | 5V |
| GND | GND |
| SCL | A5 |
| SDA | A4 |
| MQ135 | |
| VCC | 5V |
| GND | GND |
| AO | A0 |
| DO | - |
| LM393 MIC | |
| VCC | 5V |
| GND | GND |
| OUT | A1 |
| NRF24L01 | |
| VCC | 3V3 |
| GND | GND |
| CSN | D8 |
| CE | D7 |
| SCK | D13 |
| MOSI | D11 |
| MISO | D12 |

3.5. Algoritam senzorske jedinice

```
//UKLJUČITI POTREBNE BIBLIOTEKE
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <BH1750.h>
#include "MQ135.h"
#include <SPI.h>
#include "RF24.h"

//DEFINIRATI KONSTANTE I PINOVE
#define SEALEVELPRESSURE_HPA (1013.25)
const int soundPin = A1;
const int MQ135PIN=0;

//DEKLARIRATI OBJEKTE
RF24 myRadio (7, 8); // "myRadio" JE OBJEKT TIPA RF24 KORIŠTEN ZA OSTVARIVANJE KOMUNIKACIJE
Adafruit_BME280 bme; // OBJEKT SENZORA BME280
BH1750 lightMeter; // OBJEKT SENZORA BH1750
MQ135 gasSensor = MQ135(MQ135PIN); //OBJEKT SENZORA MQ135

//DEKLARIRATI VARIJABLE
byte addresses[][6] = {"Node0", "Node1", "Node2", "Node3", "Node4"}; // ADRESA SVAKE CIJEVI ZA SLUŠANJE

struct dataTransmitted{
  char a[4];
  float temp;
  float pres;
  float alt;
  float hum;
  float lux;
  int sound;
  int ppm;
};

dataTransmitted data={ "S01",0,0,0,0,0,0,0};
int count=0;
bool ok;

void setup() //SETUP FUNKCIJA, OBAVLJA SE SAMO JEDNOM PRI POKRETANJU
{
  myRadio.begin(); // POKRENI NRF24L01 RADIO
  myRadio.setChannel(108); // 108 JE KANAL IZNAD VEĆINE WIFI KANALA
  myRadio.setPALevel(RF24_PA_MAX); // MAKSIMALNA SNAGA
  myRadio.setDataRate(RF24_2MBPS); // MAKSIMALNA BRZINA PRIJENOSA PODATAKA

  myRadio.openWritingPipe(addresses[1]); // VRIJEDNOST POLJE ADDRESSES[1], ADRESA NA KOJU SE PIŠU
  PODATCI

  bme.begin(0x76); // BME280
  Wire.begin(); //POTREBNO ZA I2C
  lightMeter.begin(); // BH1750
  delay(1000); //DELAY OD 1S PRIJE LOOP FUNKCIJE
} //KRAJ SETUP FUNKCIJE

void loop() //BESKONAČNA LOOP FUNKCIJA
{
  CollectData(); //POZIVA FUNKCIJU KOJA OČITAVA I SPREMA VRIJEDNOSTI SENZORA

  do{
    count++;
    ok=myRadio.write( &data, sizeof(data) ); // POŠALJI IZMJERENE PODATKE SVE DOK

    delay(100); // VARIJABLA OK NIJE TRUE ILI ODUSTANI NAKON 15 POKUŠAJA
  }while(count<=15 && ok==false);

  count=0;
  delay(5000); // RESETIRAJ BROJ POKUŠAJA, MIRUJ 5 SEKUNDI
} //KRAJ LOOP FUNKCIJE

void CollectData(){ // FUNKCIJA KOJA SPREMA IZMJERENE VRIJEDNOSTI
  SENZORA // U STRUKTURU VARIJABLE DATA
  data.temp=bme.readTemperature(); // U STRUKTURU VARIJABLE DATA
  data.pres=bme.readPressure() / 100.0F;
  data.alt=bme.readAltitude(SEALEVELPRESSURE_HPA);
  data.hum=bme.readHumidity();
  data.lux=lightMeter.readLightLevel();
  data.sound=analogRead (soundPin);
  data.ppm=gasSensor.getPPM();
}
```

3.6. Algoritam centralne Arduino jedinice

```
//POTREBNE BIBLIOTEKE
#include <SPI.h>
#include "RF24.h"

//DEKLARIRATI KONSTANTE
long previousMillis = 0;
const long interval = 20000;

// DEKLARIRATI OBJEKTE I VARIJAVLE
RF24 myRadio (7, 8); // INSTANCA RADIJA, SPECIFIČNI CE I CS PIN

byte addresses[][6] = {"Node0", "Node1", "Node2", "Node3", "Node4"}; // ADRESE CIJEVI ZA SLUŠANJE

struct dataReceived{
  char a[4];
  float temp;
  float pres;
  float alt;
  float hum;
  float lux;
  int sound;
  int ppm;
};

dataReceived recdata; // VARIJABLA U KOJU ĆE BITI SPREMLJENI PRISTIGLI PODATCI
dataReceived data[5]; // POLJE STRUKTURA, SLUŽI ZA SPREMANJE PODATAKA IZ SENZORSKIH JEDINICA

void setup() //SETUP FUNKCIJA, OBAVLJA SE SAMO JEDNOM PRILIKOM POKRETANJA
{
  Serial.begin(115200); //INICIJALIZIRAJ SERIAL MONITOR, BRZINA 115200
  myRadio.begin();
  myRadio.setChannel(100);
  myRadio.setPALevel(RF24_PA_MAX);
  myRadio.setDataRate(RF24_2MBPS);

  myRadio.openReadingPipe(1, addresses[1]);
  myRadio.openReadingPipe(2, addresses[2]);
  myRadio.openReadingPipe(3, addresses[3]);
  myRadio.openReadingPipe(4, addresses[4]); // POSTAVI ADRESE A SLUŠANJE SVAKE SENZORSKE JEDINICE
  myRadio.startListening();
} //KRAJ SETUP FUNKCIJE

void loop() //LOOP BESKONAČNA FUNKCIJA
{
  if ( myRadio.available() ) // PROVJERI IMA LI DOSTUPNIH PODATAKA ZA PRIMANJE
  {
    while (myRadio.available() ) // SVE DOK IMA
    {
      myRadio.read( &recdata, sizeof(recdata) ); // PRIMI PODATKE I SPREMI U RECDATA VARIJABLU
    }
    int x=recdata.a[2]-'0'; // RASPOZNAJ IZ KOJE SENZORSKE JEDINICE SU STIGLI PODATCI
    strcpy(data[x].a, recdata.a); // TE IH SPREMI U ODGOVARAJUĆE POLJE STRUKTURA
    data[x].temp=recdata.temp;
    data[x].pres=recdata.pres;
    data[x].alt=recdata.alt;
    data[x].hum=recdata.hum;
    data[x].lux=recdata.lux;
    data[x].sound=recdata.sound;
    data[x].ppm=recdata.ppm;
  }

  unsigned long currentMillis = millis(); // OČITAJ PROTEKLO VRIJEME OD POKRETANA PROGRAMA
  if(currentMillis - previousMillis > interval) { //AKO JE PROŠAO ZADANI INTERVAL POKRENI SENDDATATORAS
  FUNKCIJU
  SendDatatoRas();
  previousMillis = currentMillis; //RESETIRAJ VREMENSKI INTERVAL
  }
}

void SendDatatoRas(){
  for(int i=1; i<=4; i++){ //ŠALJE PODATKE SVAKE SENZORSKE JEDINICE SERIJSKIM PUTEM LINIJU PO
  LINIJU
  Serial.println(data[i].a);
  Serial.println(data[i].temp);
  Serial.println(data[i].pres);
  Serial.println(data[i].alt);
  Serial.println(data[i].hum);
  Serial.println(data[i].lux);
  Serial.println(data[i].sound);
  Serial.println(data[i].ppm);
  }
}
```


3.7. Algoritam centralne Raspberry Pi jedinice

```
import serial
import time
import sys
import urllib.request
import time

myAPI01='HLUYNU6P30Y7AAGT'
myAPI02='DN97TYOEK0ZGWHMV'
myAPI03='R9SVEE9M0ZLG2XOZ'
myAPI04='K0Q8BVRMP9BPBE7G' //DEKLARIRAJ API VRIJEDNOSTI KANALA NA THINGSPEAK

baseURL01 = 'https://api.thingspeak.com/update?api_key=%s' % myAPI01
baseURL02 = 'https://api.thingspeak.com/update?api_key=%s' % myAPI02
baseURL03 = 'https://api.thingspeak.com/update?api_key=%s' % myAPI03
baseURL04 = 'https://api.thingspeak.com/update?api_key=%s' % myAPI04 //POSTAVI URL SVAKOG KANALA

ser=serial.Serial("/dev/ttyUSB0",115200) //INICIJALIZIRAJ SERIJSKU KOMUNIKACIJU
ser.baudrate=115200

data01=[1,0,0,0,0,0,0,0]
data02=[2,0,0,0,0,0,0,0]
data03=[3,0,0,0,0,0,0,0]
data04=[4,0,0,0,0,0,0,0] //VARIJABLE ZA SPREMANJE PRISTIGLIH PODATAKA

while True: //BESKONAČNA PETLJA DA BI PROGRAM RADIO BEZ PREKIDA
    if(ser.in_waiting>0): //PROVJEI IMA LI SERIJSKIH PODATAKA NA ČEKANJU
        line=ser.readline().strip() //SPREMI CIJELU PRISTIGLU LINIJU

        if(line==b'S01'): //DETEKTIRAJ DALI JE TO PRVA LINIJA SA S01 KAKO
            k=1
            elif(k<8 and k>=0):
                data01[k]=float(line) //PRIMI SVE PODATKE OD SENZORSKE JEDINICE S01
                k=k+1

            if(k==8):
                k=9 //AKO JE KRAJ PODATAKA S01, POŠALJI PODATKE NA THINGSPEAK, TE ČEKAJ S02
                //PONOVNI PROCES ZA SVAKU SENZORSKU JEDINICU TE NAKON S04 RESETIRAJ BROJAC
                // PRIMLJENJIH LINIJA, T.J. PODATAKA
                conn = urllib.request.urlopen(baseURL01 +
                '&field1=%s&field2=%s&field3=%s&field4=%s&field5=%s&field6=%s&field7=%s' % (data01[1],
                data01[2],data01[3],data01[4],data01[5],data01[6],data01[7]))
                conn.close()

            if(line==b'S02'):
                k=9
                elif(k>8 and k<=15):
                    data02[k-8]=float(line)
                    k=k+1

            if(k==16):
                k=17
                conn = urllib.request.urlopen(baseURL02 +
                '&field1=%s&field2=%s&field3=%s&field4=%s&field5=%s&field6=%s&field7=%s' % (data02[1],
                data02[2],data02[3],data02[4],data02[5],data02[6],data02[7]))
                conn.close()

            if(line==b'S03'):
                k=17
                elif(k>16 and k<=23):
                    data03[k-16]=float(line)
                    k=k+1

            if(k==24):
                k=25
                conn = urllib.request.urlopen(baseURL03 +
                '&field1=%s&field2=%s&field3=%s&field4=%s&field5=%s&field6=%s&field7=%s' % (data03[1],
                data03[2],data03[3],data03[4],data03[5],data03[6],data03[7]))
                conn.close()

            if(line==b'S04'):
                k=25
                elif(k>24 and k<=31):
                    data04[k-24]=float(line)
                    k=k+1

            if(k==32):
                k=-1
                conn = urllib.request.urlopen(baseURL04 +
                '&field1=%s&field2=%s&field3=%s&field4=%s&field5=%s&field6=%s&field7=%s' % (data04[1],
                data04[2],data04[3],data04[4],data04[5],data04[6],data04[7]))
                conn.close()
```

POPIS SLIKA

Slika 2.1. Komponente kondenzatorskog mikrofona

<http://www.mxlmics.com/blog/how-condenser-microphones-work/>

Slika 2.2. Intenzitet svjetla na površini sfere

<https://e-radionica.com/hr/blog/2015/08/19/vrste-i-karakteristike-svjetla/>

Slika 2.3. Oznaka za fotodiodu

<https://hr.wikipedia.org/wiki/Fotodioda>

Slika 2.4. Prikaz temperature i vlage pomoću ThingSpeak platforme

<https://thingspeak.com/channels/1122948>

Slika 3.1. Arduino Nano

<https://store.arduino.cc/arduino-nano>

Slika 3.2. Raspberry Pi 3 model B+

<https://www.chipoteka.hr/artikl/139241/raspberry-pi-3-model-b-wifi-bluetooth-onboard-8208000133>

Slika 3.3. BME280 modul povezan na Arduino Uno

<https://randomnerdtutorials.com/bme280-sensor-arduino-pressure-temperature-humidity/>

Slika 3.4. Blok dijagram IC-a

<https://e-radionica.com/hr/blog/2015/10/21/kkm-bh1750/>

Slika 3.5. LM393 zvučni senzor modul

<https://arduino.stackexchange.com/questions/20266/how-do-i-use-sound-sensor-lm393-speaker-in-arduino>

Slika 3.6. MQ135 senzor

<https://ardubotics.eu/hr/senzori/1103-mq135-sensor-air-quality-sensor-hazardous-gas-detection-module-arduino.html>

Slika 3.7. Konektori nRF24L01 modula

<https://e-radionica.com/hr/blog/2015/10/27/kkm-nrf24l01/>

Slika 3.8. 7805 regulator napona

<https://www.iotwebplanet.com/product/7805-5v-voltage-regulator/>