Razvoj i testiranje senzorskog čvora za analizu vlažnosti tla

Kokošarević, Hrvoje

Master's thesis / Diplomski rad

2019

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://urn.nsk.hr/urn:nbn:hr:200:673531

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-14**

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

RAZVOJ I TESTIRANJE SENZORSKOG ČVORA ZA ANALIZU VLAŽNOSTI TLA

Diplomski rad

Hrvoje Kokošarević

Osijek, 2019.

SADRŽAJ:

1. UVOD **1**
2. OPIS KORIŠTENIH TEHNOLOGIJA **3**
2.1. Internet objekata (Internet of Things) **3**
2.2. LoRa **4**
2.3. ThingSpeak **7**
2.4. Mikro upravljači **9**
2.5. Senzori **11**
2.6. Problemi sa senzorima **16**
3. OPIS KODA **17**
3.1. Problemi sa ThingSpeak poslužiteljem **23**
4. ANALIZA REZULTATA **25**
5. ZAKLJUČAK **40**
6. POPIS FORMULA **41**
7. LITERATURA **42**
8. SAŽETAK **44**
9. NASLOV I SAŽETAK NA ENGLESKOM JEZIKU **45**
10. ŽIVOTOPIS **46**
11. PRILOZI **47**

1. UVOD

U brojnim situacijama javlja se potreba za mjerenjem vlažnosti tla i kontinuiranim nadzorom spomenutih vrijednosti. Poljoprivrednici koji prate vlažnost tla na svojim poljima mogu maksimizirati urod planiranim natapanjem i navodnjavanjem tla. Mjerenje i nadzor točne količine vlage u zemlji pomaže poljoprivrednicima u održavanju zdravlja zemlje, pomaže pri štednji vode za natapanje te povećanju uroda. Prema [1] predviđa se kako bi generirani podaci o vlažnosti zemlje mogli poljoprivrednicima smanjiti potrošnju vode i do 30%. S druge strane u građevinarstvu se također mogu vidjeti prednosti kontinuiranog nadzora vlažnosti tla. Ukoliko je na nekoj lokaciji na kojoj se planira gradnja, tlo u jednom dijelu godine previše vlažno odnosno rahlo, to može značiti loše uvjete za postavljanje temelja za gradnju na toj lokaciji. Ovakvim preventivnim djelovanjem može se između ostaloga, izbjeći i moguće naknadno nanošenje štete ili u krajnjem slučaju urušavanje građevine, odnosno izbjegava se katastrofa. Također ukoliko se na vrijeme na postojećoj građevini uvide problematične razine vlažnosti i sastava tla, može se na vrijeme reagirati i spriječiti urušavanje. Razvoj ovakvog uređaja koji može kontinuirano pružati informacije o vlažnosti tla uvelike doprinosi krajnjim korisnicima sustava, ali i općem dobru okoline.

Internet objekata (Internet of Things) omogućava povezivanje puno manjih uređaja koji prate neke osobine iz svoje okoline sa internetom odnosno omogućava korisniku konstantnu povezanost sa uređajima u svijetu oko sebe. Ovaj rad omogućava korisniku kontinuirani nadzor vlažnosti tla na svojem posjedu ili drugoj, udaljenoj lokaciji, korištenjem uređaja razvijenim u sklopu ovoga rada.

Često je potrebno biti povezan na velikim udaljenostima gdje nema pokrivanja mobilnom mrežom. Tu uvelike može pomoći LoRa WAN (engl. *Long Range Wireless Access Network*) koja omogućava povezivanje IoT uređaja na velikim udaljenostima bez potrebe za izgradnjom skupe infrastrukture. LoRa omogućuje prijenos male količine podataka na udaljenostima do nekoliko kilometara što je sasvim dovoljno za ovakvu primjenu.

Postoji više rješenja koja omogućavaju prikupljanje podataka sa senzora za vlagu te spremanje, analizu i vizualizaciju istih. Neka od rješenja su „DIY“ - uradi sam (engl. *do it yourself*) verzije spomenutog koja koriste mikro upravljače poput Arduina, module za LoRa komunikaciju itd. S druge strane postoje i komercijalna rješenja poput onog tvrtke Kerlink koje omogućava komunikaciju senzora za vlagu sa internetom, odnosno sa poslužiteljem na kojem se može pratiti trenutno stanje vlage u zemlji [2].

Rad je podijeljen u pet dijelova; 1) opis korištenih tehnologija – gdje će biti objašnjeno zašto se i kako koristi koja od tehnologija, te opis razvijenog sklopa senzora i uređaja; 2) opis koda – kratak opis napisanog računalnog programa; 3) analiza rezultata – bit će uspoređeni rezultati mjerenja raznih vrsta zemlje sa raznih lokacija, te će biti uspoređeni korišteni senzori; biti će ustanovljeno koji je od senzora najbolji za mjerenje te u kojim se primjenama može koristiti; te 4) zaključak.

2. OPIS KORIŠTENIH TEHNOLOGIJA

2.1.Internet objekata (engl. *Internet of Things*)

Internet objekata sastoji se od velikog broja uređaja koji mogu osjetiti, komunicirati, računati i ukoliko je omogućeno, aktivirati druge uređaje. Internet objekata uključuje pristupanje, nadzor, komunikaciju i kontrolu raznih senzora i uređaja putem interneta. Ključni pojam jest komunikacija između uređaja.

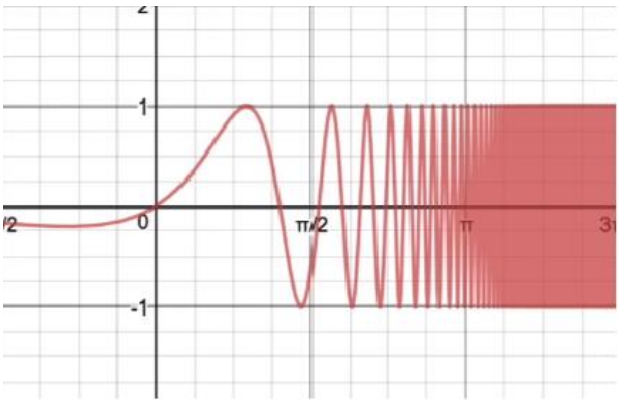
U svrhu boljeg objašnjenja Interneta objekata, naveden je primjer. Pametne kuće predstavljaju sam vrhunac interneta objekata, jer uključuju najveći broj povezanih uređaja. Kućni aparati poput detektora dima, senzora za kvalitetu zraka, pametna zvona, uređaja za kontrolu temperature te ostali pripadajući uređaji, mogu komunicirati sa pametnim telefonima, pametnim satovima ili narukvicama, te na taj način prilagoditi okolinu korisniku. Pametna narukvica ili sat koju osoba nosi za vrijeme spavanja, može osjetiti kada se osoba nalazi u laganom snu, te može poslati obavijest pametnoj budilici da zvoni i time probudi osobu u najboljem trenutku u odnosu na trenutno stanje njegovog ciklusa sna. Pametna budilica, zajedno sa pametnim telefonom, može provjeriti vremensku prognozu za korisnikovu lokaciju te netom prije buđenja prilagoditi uvjete unutar pametne kuće na način da postavi klimatizacijski uređaj na odgovarajuću temperaturu. Isto se može učiniti i u automobilu. Pregledom vremenske prognoze aplikacija za navigaciju na pametnom telefonu može predvidjeti mogućnost pojavljivanja gužvi u prometu te na taj način predvidjeti u kolikoj će mjeri vremenske prilike utjecati na uvjete u prometu. Samim time, navigacijska aplikacija promijeniti će rutu do posla koju korisnik često koristi [3]. Na ovaj način Internet objekata može uvelike pridonijeti automatizaciji nekih aspekata života korisnika, a samim time i unapređenju njegova života.

2.2. LoRa

LoRa (engl. *Long Range*) je modulacijska tehnika koja omogućava prijenos podataka na velikim udaljenostima na relativno niskim prijenosnim brzinama na uskom dijelu spektra [4]. Kako se radi o bežičnom načinu komunikacije, LoRa tehnika definira način na koji se podaci prenose kroz zrak. Poseban hardver (engl. *hardware*) je potreban kako bi se LoRa komunikacija odvijala između uređaja [5]. LoRa je svoju primjenu pronašla u pametnoj infrastrukturi, pametnim vozilima te industrijskoj kontroli i nadzoru [6].

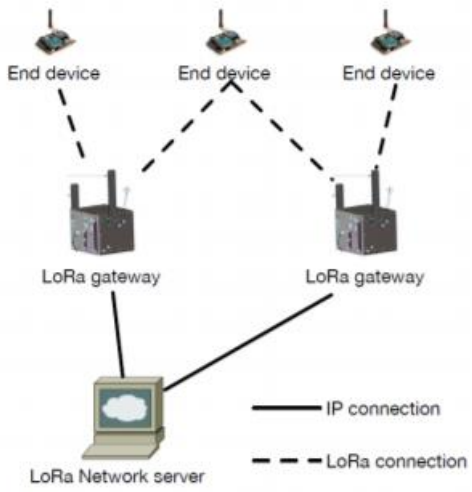
Prema Semtech web stranici [7] LoRa koristi tehniku kojom se podaci (spektar signala) prije prijenosa raspršuju na strani pošiljatelja, te se na prijemnoj strani nakon prijema sažimaju. Na taj način se osigurava sigurnost podataka, uz nisku potrošnju energije. LoRaWAN je prema [7], protokol mreže širokog područja (engl. *Wide Area Network*) niske potrošnje energije. Kreiran je s ciljem bežičnog povezivanja mnoštva uređaja niske potrošnje energije sa internetom, čineći tako Internet objekata (engl. *Internet of Things*). Specifikacija definira parametre koji se koriste za komunikaciju između uređaja i infrastrukture na fizičkom sloju. LoRa modul koji se koristi prilikom izrade ovog rada napravila je tvrtka Semtech, dok LoRa Alliance® [8] neprofitna organizacija, pruža podršku i pokreće standardizaciju te globalnu harmonizaciju LoRaWAN protokola.

LoRa koristi modulacijsku tehniku (engl. *chirp spread spectrum*) u kojoj je jedan simbol (nula ili jedinica) predstavljen ili rastom ili padom frekvencije sinusnog signala u vremenu. Često je rast ili pad opisan određenom funkcijom koja povezuje promjenu frekvencije u vremenu. Funkcija koja opisuje promjenu signala može biti npr. linearna ili bilo koja druga funkcija [9]. Primjer simbola koji predstavlja jedinicu u modulacijskoj tehnici koju LoRa koristi vidljiv je na slici 2.1.



Slika 2.1. – Prikaz „chirp spread spectrum“ signala [10]

Tipična LoRa mreža ima zvijezda topologiju [6], koja uključuje tri različite vrste mrežnih uređaja; LoRa poslužitelj, LoRa mrežni prolaz (engl. *gateway*) te LoRa uređaj koji pristupa mreži (engl. *end device*), vidljivo sa slike 2.2.

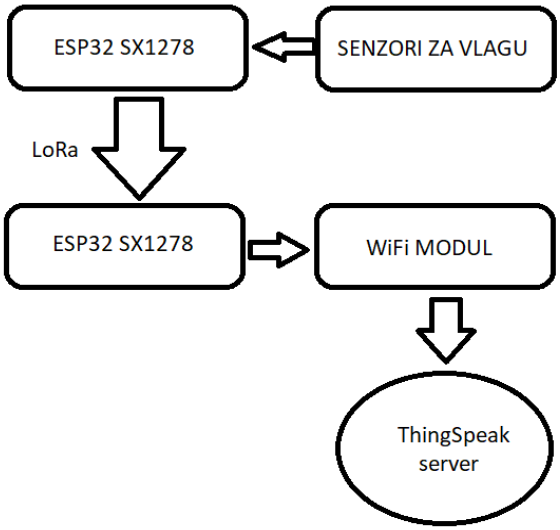


Slika 2.2. – Prikaz LoRa mrežne arhitekture [6]

Jedan LoRa mrežni prolaz (engl. *gateway*) može posluživati šire područje grada od nekoliko stotina četvornih kilometara. No, veličina tog područja je jako ovisna o vremenskim uvjetima te o sastavu terena (brdovito, šumovito, visoke zgrade...) [5]. LoRa često predstavlja samo zadnji čvor pristupne mreže za uređaje koji se nalaze na udaljenim lokacijama, odnosno ne koristi se cjelokupna mreža sa sva tri mrežna uređaja.

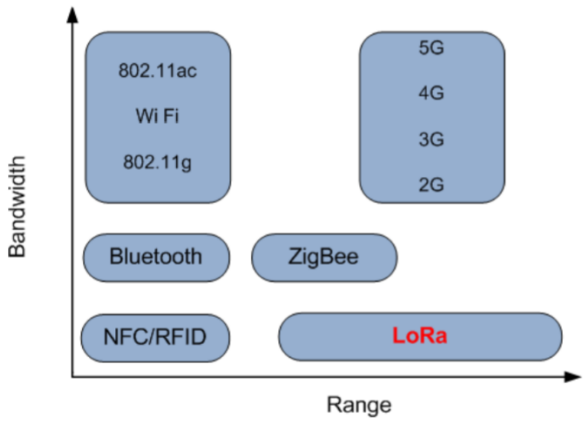
Osim korištenja LoRa-e na navedenom primjeru korištenjem gateway-a i LoRa network servera, LoRa se može koristiti i samo kao zadnji korak u komunikaciji sa internetom. U ovom radu LoRa je korištena upravo na taj način. Jedan LoRa modul je povezan sa uređajem koji čita vrijednosti sa senzora, te ih šalje drugom uređaju koji prima LoRa podatke. Drugi uređaj koji je spojen na internet ih tada šalje na poslužitelj na kojem se podaci spremaju i prikazuju. Drugi uređaj je na Internet spojen korištenjem WiFi modula. Ovakvim načinom komunikacije ne koristi se cjelokupna LoRa mreža nego samo LoRa tehnologija koji omogućuje komunikaciju između dva krajnja uređaja.

Na ovaj način LoRa predstavlja samo zadnji korak u komunikaciji između interneta i objekata. Model komunikacije korišten u ovom radu nalazi se na slici 2.3.



Slika 2.3. – Prikaz toka podataka kroz sustav

LoRa koristi relativno niske nelicencirane frekvencije; u Europi su to 169MHz, 433MHz, 868MHz, dok je u Sjevernoj Americi to 915MHz [4]. U ovom radu korištena je frekvencija od 433MHz-a. Ono što karakterizira LoRa tehnologiju jest velik domet na relativno uskom frekvencijskom spektru. Prikaz odnosa spektra i dometa LoRa tehnologije usporedno sa drugim bežičnim tehnologijama vidljiv je na slici 2.4.



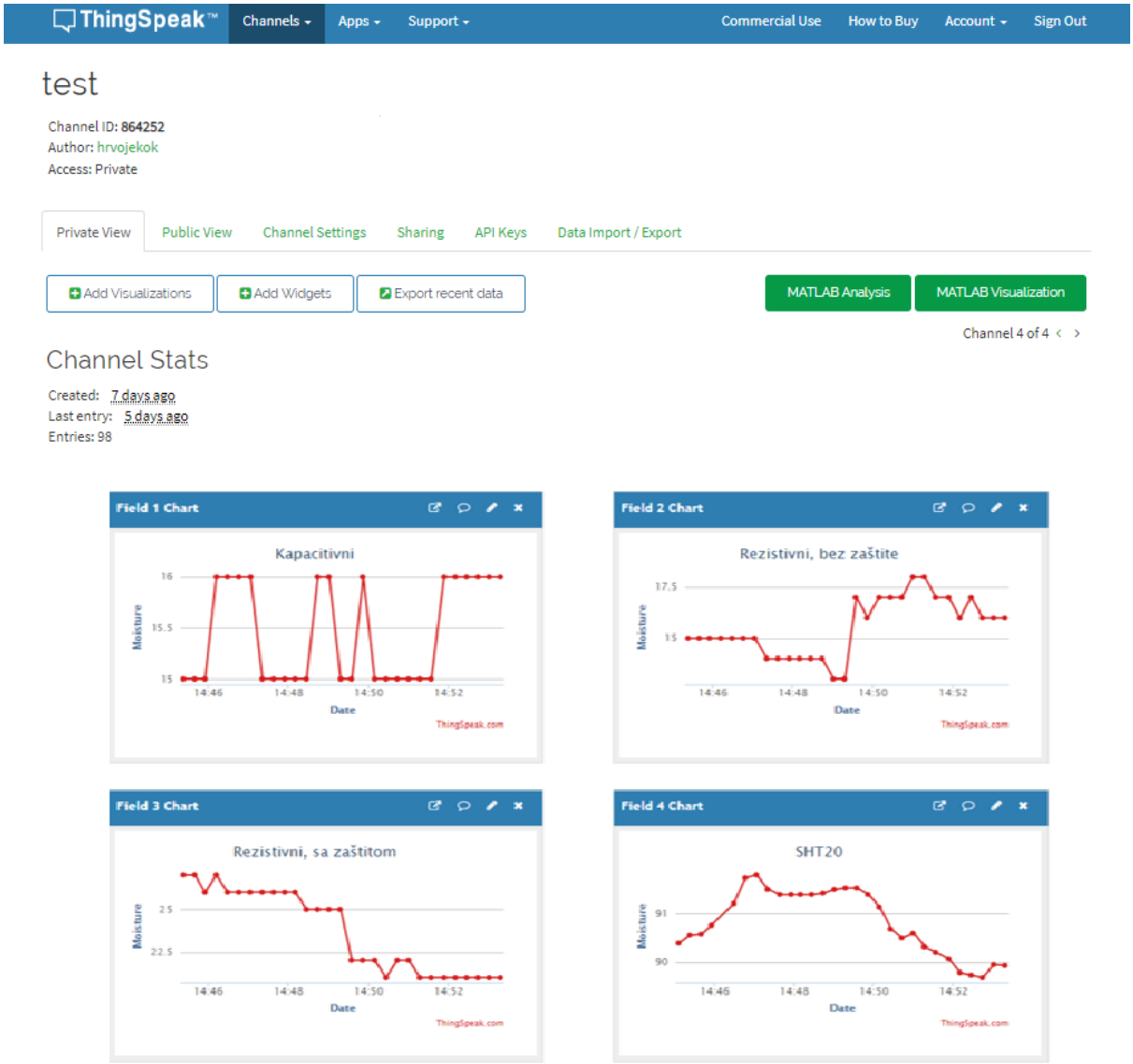
Slika 2.4. – Prikaz odnosa spektra i dometa LoRa tehnologije u odnosu na ostale bežične tehnologije [4]

Valna duljina vala je relativno mala, pa nije potrebna velika količina energije odašiljaču kako bi se postigla pokrivenost na većim udaljenostima. Brzine prijenosa podataka koje LoRa može uspostaviti su niske, a nalaze se u rasponu između 11kbit/s i 250kbit/s [11].

2.3. ThingSpeak

Svi podaci koji dolaze sa senzora putem mikroprocesorskih upravljača moraju negdje biti spremljeni kako bi se izvršila daljnja obrada, analiza i prikazivanje podataka. Svrha prikupljanja podataka sa senzora jest analizirati ih, obraditi i na kraju omogućiti korisniku pristup i vizualizaciju podataka sa senzora na bilo kojem mjestu te u bilo kojem trenutku. Kako jako velik broj ljudi na svijetu ima omogućen pristup internetu, bilo je potrebno podatke spremiti na nekakav javno dostupni poslužitelj. S druge strane, potrebno je omogućiti pristup podacima samo autoriziranim korisnicima. Postoji više načina na koji se ovo može riješiti. Jedan od načina je izrada vlastite web stranice na koju bi se podaci spremali, prikazivali i analizirali. Zbog toga što već postoje gotova rješenja za prikupljanje, spremanje, analizu i prikaz podataka sa senzora, korišteno je upravo jedno takvo rješenje. Osim korištene ThinsSpeak platforme, postoje razne druge, kao što su Thingsboard ili ThingWorx. Od svih dostupnih platformi ThingSpeak je odabran iz razloga što se činio jednostavnim za implementaciju i korištenje [12]. ThingSpeak je IoT (engl. *Interner of Things*) analitička platforma, odnosno usluga koja dozvoljava korisniku skupljanje, vizualizaciju te analizu podataka sa senzora u stvarnom vremenu. ThingSpeak pruža trenutnu vizualizaciju podataka poslanih sa IoT uređaja. Omogućena je i izvedba MATLAB® koda direktno unutar ThingSpeak web stranice. To omogućava razne načine obrade podataka koji se nalaze na poslužitelju. Sve to omogućava da se ThingSpeak koristi za razne vrste prototipiranja i dokazivanja koncepta (engl. *proof of concept*), bez potrebe za izradom posebne web stranice. Na taj način skraćuje se vrijeme i trošak izrade projekta.

ThingSpeak web sučelje omogućuje kreiranje kanala, te polja za unos podataka. Svako polje predstavlja podatke koji dolaze sa jednog senzora. Izgled ThingSpeak sučelja prikazan je na slici 2.5. Postoji velik broj mogućnosti za manipuliranje podacima na ThingSpeak poslužitelju [12].



Slika 2.5. – Izgled ThingSpeak sučelja [12]

2.4. Mikro upravljači

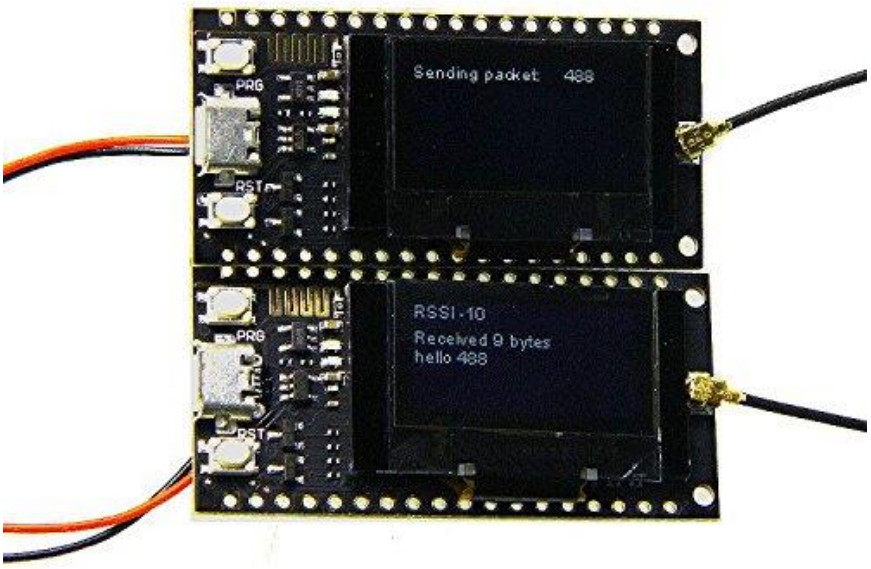
Uređaj koji je razvijen u sklopu ovog rada sastoji se od više komponenti. Prva komponenta jest skup senzora koji služe za očitavanje vlage tla. Idući su mikoprocesorski upravljači koji služe za upravljanje senzorima, te dodatni moduli koji služe za razne oblike komunikacije između mikroprocesorskih upravljača, te između upravljača i interneta. Kako bi bilo pokriveno što više vrsta senzora korištena su 4 senzora, od kojih su dva rezistorska, jedan kapacitivni, te posljednji koji je također kapacitivni, ali koristi I^2C sabirnicu za komunikaciju sa mikro upravljačem.

Postoje razni mikroprocesorski uređaji te prikladni moduli pomoću kojih je moguće ostvariti komunikaciju između senzora i interneta. Kako je zahtjev rada bio ostvariti komunikaciju senzora koji se nalaze na udaljenom mjestu gdje nema pristupa internetu, bilo je potrebno ostvariti komunikaciju udaljenog uređaja sa internetom. Kao jedno od rješenja bilo bi korištenje modula sa SIM karticom, koji bi koristio mobilne podatke kao i bilo koji pametni telefon. Takav uređaj ima nekoliko potencijalnih problema. Prvo, on zahtjeva mjesečnu naknadu kod operatera mobilnih usluga, samim time je skuplji za korištenje. Drugo, postoji problem pokrivanja lokacije uređaja signalom mobilne mreže. Postoji velika mogućnost da mjesto na kojem se taj uređaj nalazi neće biti pokriveno mobilnom mrežom. Serijska komunikacija korištenjem kablova (poveznih žica) nije dolazila u obzir jer se radi o udaljenostima od nekoliko kilometara.

Kao odlično rješenje istaknula se LoRaWAN komunikacija. Kao što je već objašnjeno LoRaWAN (engl. *Long Range Wide Area Network*) jest bežična tehnologija koja radi na frekvencijama ispod 1GHz-a, te omogućuje prijenos male količine podataka na relativno velikim udaljenostima.

Postoje razni LoRa moduli koji se mogu spojiti sa nekim od mikroprocesorskih upravljača korištenjem serijske komunikacije, no kao najprikladniji pokazao se SX1278 koji je direktno spojen na istoj pločici sa ESP32 mikro upravljačem. ESP32 je serija jeftinih SoC (engl. *System on Chip*) uređaja niske potrošnje, koji na sebi imaju integriran Wi-Fi modul, Bluetooth modul, omogućenu serijsku komunikaciju korištenjem I^2C sučelja itd. Osim toga, imaju dvojezgrene procesore te 520kB RAM memorije. Sve to čini ESP32 seriju mikroprocesorskih upravljača iznimno snažnim i sposobnim za primjenu u domeni Interneta objekata (engl. *Internet of Things*) – u odnosu na druge mikroprocesorske upravljače kao na primjer Arduino Uno.

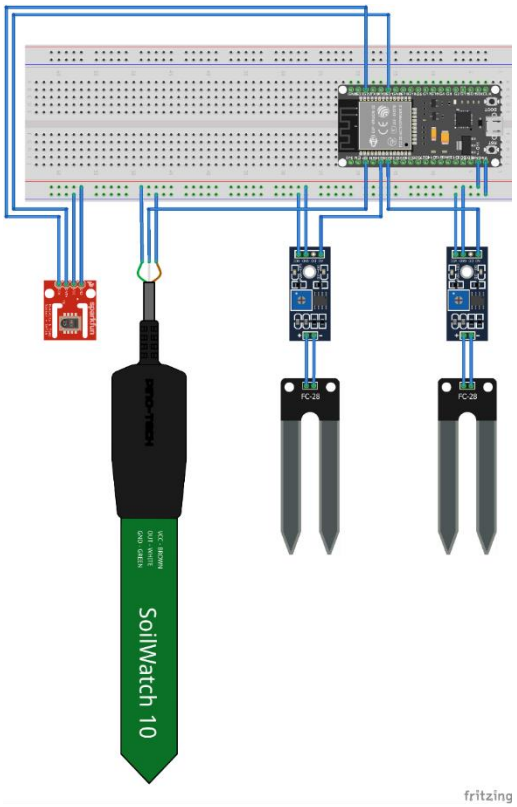
Iz navedenih razloga kao najprikladniji upravljač odabran je ESP32 sa integriranim SX1278 LoRa modulom. To je značilo pojednostavljenje cijelog sustava utoliko što nije bilo potrebno dodatno spajati LoRa modul sa mikro upravljačem, što u konačnici rezultira smanjivanjem mogućnosti hardverske pogreške. Osim što ESP32 na sebi ima LoRa modul, također dolazi i sa ugrađenim Wi-Fi modulom, pomoću kojega se, korištenjem prikladne Wi-Fi biblioteke može spojiti na bilo koju Wi-Fi mrežu, što olakšava ostvarivanje komunikacije sa internetom. Mikroprocesorski upravljač koji je korišten prilikom izrade na sebi također ima i 0.96 inch-ni OLED ekran, koji omogućuje lakše prikazivanje podataka koji se šalju i primaju prilikom komunikacije. Također omogućava i lakše otklanjanje pogrešaka. Mikroprocesorski upravljač korišten prilikom izrade ovog rada nalazi se na slici 2.6.



Slika 2.6. – Mikro upravljač ESP32 SX1278 [13]

2.5. Senzori

Senzorski čvor se sastoji od dva glavna dijela: mikro upravljača koji očitava podatke sa senzora i šalje na drugi mikro upravljač, te drugi mikro upravljač koji te podatke prima i šalje na poslužitelj. Shema mikro upravljača na kojeg su spojeni senzori nalazi se na slici 2.7.

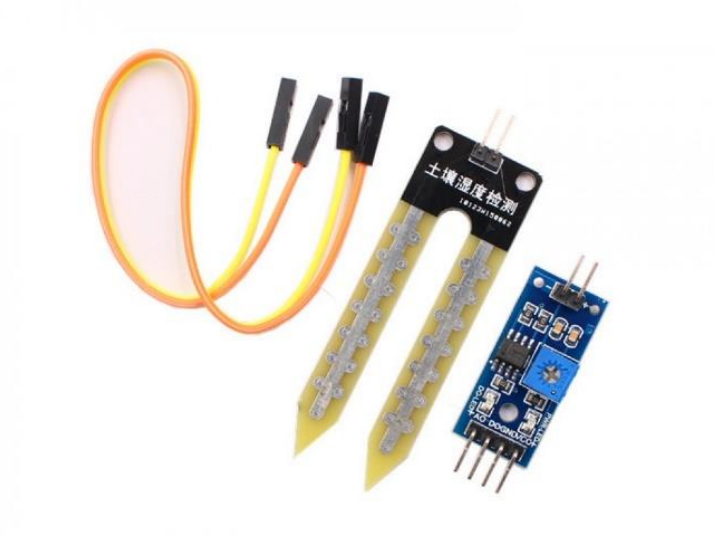


Slika 2.7. – Fritzing model senzorskog čvora

Korištena su 4 senzora, svaki različitog načina rada, kvalitete izrade, cijene, preciznosti i pouzdanosti.

Korištena su dva rezistivna senzora, jedan kapacitivni, te SHT20, također kapacitivni, ali on koristi I^2C komunikaciju, za razliku od običnog kapacitivnog koji se direktno spaja na mikro upravljač. Prva dva senzora rade na istom načelu. Radi se o rezistivnim senzorima, koji mjere otpor zemlje između dvije elektrode senzora.

Na slici 2.8. nalazi se prvi rezistivni senzor bez dodatne zaštite. On se sastoji od dva dijela. Prvi dio su elektrode koje se uranjaju u zemlju. Drugi dio je integrirani strujni krug (IC- engl. *Integrated Circuit*) na kojem se nalazi komparator LM393. Postoje dva izlaza sa senzorskog sklopa, samim time i dva načina rada; analogni i digitalni [14]. U ovom radu je korišten analogni način rada.



Slika 2.8. – Rezistivni senzor bez zaštite [15]

Senzor je predstavljen kao otpornik. Struja koja prolazi između elektroda senzora daje napon na tom senzoru koji je predstavljen otporom. Napon na senzoru jest napon na analognom izlazu na pomoćnom integriranom strujnom krugu. U kontekstu rada, kada se nalazi veća količina vlažnosti u zemlji, zemlja omogućuje provođenje veće količine struje kroz elektrode senzora. Veća količina struje kroz senzor daje veći napon na predstavljenom otporu, samim time i veću izlaznu vrijednost na analognom izlazu integriranog kruga. Senzor mjeri struju koja prolazi između dvije stezaljke, te na osnovu izmjerene struje izračunava otpor tvari u koje je uronjen. U kontekstu ovog rada, kada se nalazi veća količina vode u zemlji, zemlja će omogućiti provođenje veće količine struje, što znači da će u zemlji biti manji otpor. Manji otpor u zemlji znači veću vlažnost tla.

Na upravljaču se vrijednost napona preračunava u vrijednost između 0 i 4095. U programskom kodu se vrijednost od 0 do 4095 pretvara u vrijednosti postotaka, 4095 predstavlja 0% vlažnosti, dok 0 predstavlja 100% vlažnosti. U slučaju digitalnog načina rada, integrirani strujni krug uspoređuje vrijednost poznatog otpora sa promjenjivog otpornika i vrijednost otpora na stezaljkama senzora. Na temelju toga na digitalnom izlazu integriranog kruga dobiva se vrijednost 0 ili 1. Prag na kojem će 0 postati 1 se određuje mijenjanjem vrijednosti promjenjivog otpornika.

Drugi senzor je također rezistivni kao što je prije navedeno te ima identičan način rada kao i prethodni senzor. Jedina razlika je to što ima na sebi dodatnu zaštitu protiv oksidacije elektroda. Slika ovog senzora nalazi se na slici 2.9.

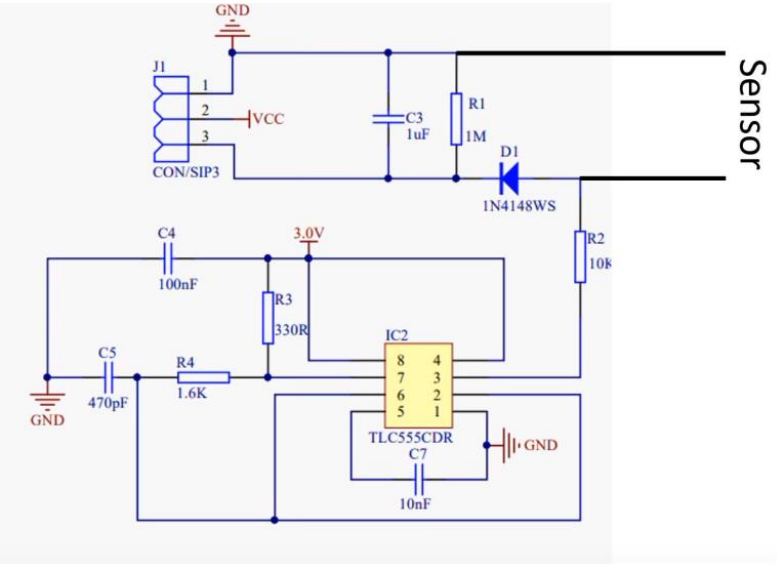
Slika 2.9. - Rezistivni senzor sa zaštitom [16]

Treći senzor je kapacitivni i on radi na potpuno drugačiji način od rezistivnih. Na slici 2.10. vidljivo je da kapacitivni senzor nema izloženih elektroda direktno u zemlji. To sprječava korodiranje senzora, što ga čini dugotrajnijim od rezistivnih senzora.

Slika 2.10. – Kapacitivni senzor [17]

Za kontrolu kapacitivni senzor koristi NE555 brojač (engl. *timer*) koji generira pravokutni signal koji se šalje na jedan pol senzora.

Shema senzora vidljiva je na slici 2.11. [18]. Drugi pol senzora povezan je sa GND – uzemljenjem. Dvije izolirane noge odnosno pola senzora definiraju kondenzator, pa napon na otporniku koji se nalazi uz kondenzator definira količinu vlage u zemlji.

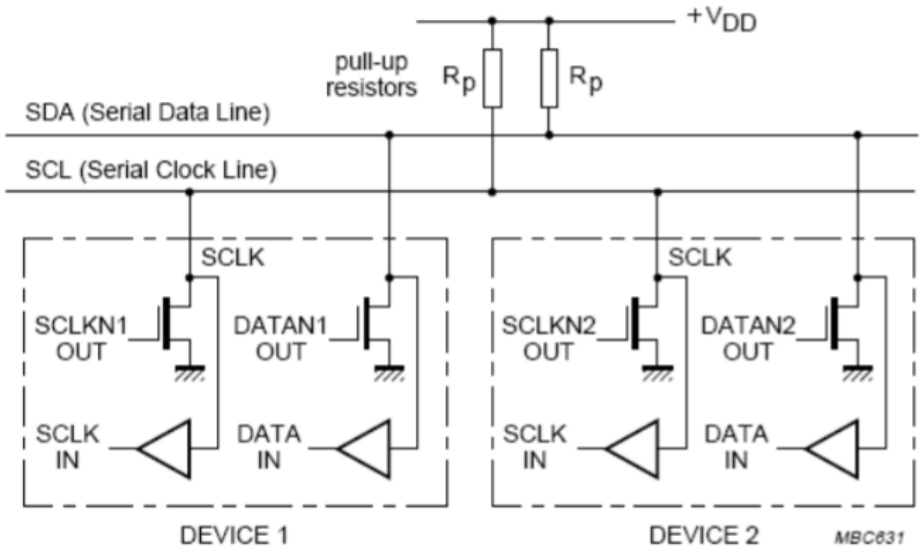


Slika 2.11. - Shema kapacitivnog senzora [18]

Posljednji senzor je SHT20. To je senzor koji uz mjerenje vlage ima i mogućnost mjerenja temperature. Jedini od korištenih senzora ima tu mogućnost. Za ovu upotrebu mjerenje temperature nije bilo potrebno, ali za neke druge primjene nije loše imati mjerenje i temperature i vlažnosti u istom senzoru. SHT20 senzor koji je korišten u ovom radu nalazi se na slici 2.12. Sa slike je vidljivo kako senzor dolazi u robusnom metalnom kućištu koje bi trebalo pružiti veliku otpornost na loše radne uvjete.

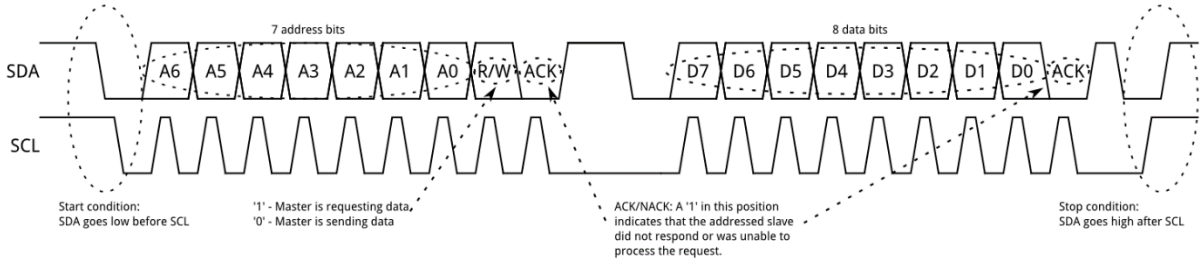
Slika 2.12. – SHT 20 senzor [19]

SHT20 je senzor koji se nalazi na integriranom krugu (IC- engl. *Integrated Circuit*). Ima četiri linije za povezivanje. Dva se koriste za napajanje (5V i GND), a druga dva predstavljaju SDA i SCL. Ovaj senzor koristi I^2C sabirnicu za komunikaciju između senzora i uređaja koji ga kontrolira [20]. I^2C sabirnica je sinkrona, serijska sabirnica koja se koristi za međusobno povezivanje raznih digitalnih i analognih sustava, među kojima su najčešći mikro upravljači, LDC pokazivači itd. Način spajanja preko I^2C sabirnice nalazi se na slici 2.13.



Slika 2.13. – Shematski prikaz I^2C sabirnice [20]

Komunikacija preko I^2C sabirnice izvodi se preko dvije linije; to su SDA i SCL. Preko SCL linije (engl. *Serial Clock*) prenosi se takt poruke, dok se preko SDA (engl. *Serial Data*) linije prenose podaci. Prikaz SDA i SCL linija I^2C sabirnice nalazi se na slici 2.14.



Slika 2.14. – Prikaz SDA i SCL linija [20]

2.6. Problemi sa senzorima

Kao što je već prije navedeno korištena su četiri senzora različitog načina rada, kvalitete izrade, cijene, preciznosti i pouzdanosti. Navedene karakteristike senzora uzrokuju različita ponašanja senzora u stvarnim radnim uvjetima. Ono što je zamijećeno jest da oba rezistivna senzora (sa zaštitom i bez nje) ne pokazuju konstantne vrijednosti vlažnosti nakon određenog vremena korištenja. Ono što je uočeno prilikom inicijalnog postavljanja senzora jest da se maksimalna i minimalna vrijednost senzora mijenjaju. Odnosno, smanjuje se radni raspon senzora. Kao primjer; ono što je nekada bilo 98% vlažnosti, sada predstavlja 100%.

Ono što je uzrok ovakvom ponašanju jest oksidacija elektroda na senzorima. U početku, na senzorima nema vidljivih tragova trošenja, odnosno oksidacije, ali dolazi do prirodnog trošenja pozitivne elektrode zbog oslobađanja elektrona prema negativnoj elektrodi. Ovakvi senzori imaju vrlo kratak rok trajanja. U samom startu, dok još nema vidljivih tragova potrošnje elektroda, ovakvi senzori pokazuju netočne i nekonzistentne rezultate, te nakon određenog vremena pojavljuju se vidljivi tragovi oksidacije. Ekscesivno trošenje pozitivne elektrode rezistivnog senzora vidljivo je na slici 2.15 [21].

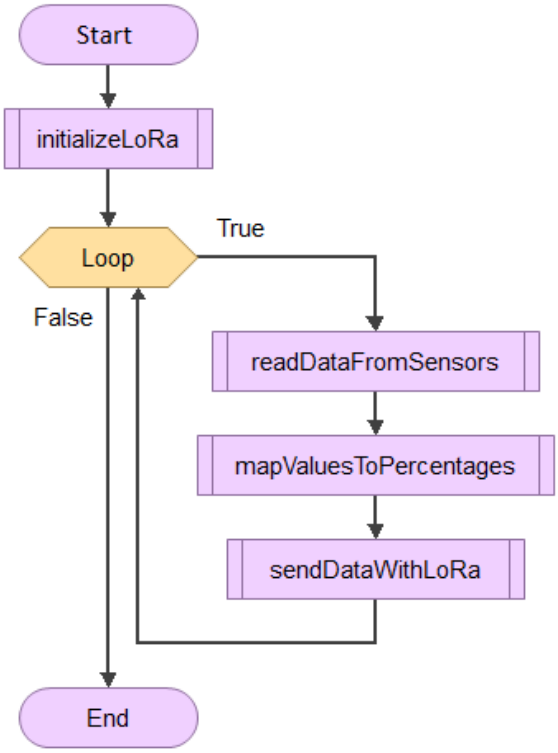
Slika 2.15. – prikaz oksidiranih elektroda rezistivnog senzora [21]

3. OPIS KODA

Tok podataka kroz sustav prikazan je na slici 2.3. Sukladno tome, vidljivo je da postoje dva mikro upravljača koji omogućuju kontrolu i komunikaciju između senzora i poslužitelja. Kako bi bio ostvaren tok podataka sa slike 2.3, potrebno je omogućiti komunikaciju između senzora i ESP32 mikro upravljača, između dva ESP32 mikro upravljača, te između ESP32 mikro upravljača i ThingSpeak poslužitelja. Komunikacija između senzora i ESP32 mikro upravljača ostvaruje se serijskom komunikacijom. Komunikacija između dva ESP32 mikro upravljača ostvaruje se korištenjem LoRa tehnologije, dok se komunikacija između ESP32 i ThingSpeak poslužitelja ostvaruje povezivanjem ESP32 mikro upravljača na Internet pomoću WiFi modula, a sama komunikacija odvija se korištenjem HTTP protokola. Iz navedenih zahtjeva za komunikaciju, može se zaključiti da je potrebno napisati dva programa koji će se izvoditi na dva odvojena ESP32 mikro upravljača.

Kao razvojno okruženje u kojem se razvija programski kod za mikro upravljač ESP32 može se koristiti Arduino IDE (engl. *Integrated Development Environment*). Kako bi se programski kod napisan u Arduino IDE razvojnom okruženju mogao izvoditi na ESP32 upravljaču potrebno je instalirati i uključiti potrebne biblioteke (engl. *library*). Navedene biblioteke omogućuju komunikaciju ESP32 upravljača sa računalom putem USB sučelja.

Kod koji upravlja mikro upravljačima sastoji se iz dva dijela; pošiljatelj (engl. *transmitter*) i primatelj (engl. *receiver*), a napisan je u C++ jeziku. Najvažniji dijelovi koda prikazani su u kasnijem dijelu teksta, dok je cijeli kod dostupan na *github* web stranici [22] odnosno u prilogu. Osnovni model odnosno tok stanja programa koji se koristi za slanje (engl. *transmitter*) može se vidjeti na slici 3.1.



Slika 3.1. – Tok dijagram programa za slanje podataka (engl. transmitter)

Sa slike 3.1. je vidljivo da se radi o programu koji jedan svoj dio mora konstantno ponavljati. U Arduino programu postoje osnovna dva dijela programa; to su setup() i loop() funkcije. U setup() funkciji izvršava se kod koji se mora izvršiti samo jednom, odnosno prilikom prvog pokretanja programa. U loop() funkciji nalazi se kod koji se mora konstantno ponovno izvršavati.

U setup() funkciji Arduino koda postavljena je funkcija koja služi za inicijalizaciju LoRa modula vidljiva u nastavku na slici 3.2.

```
void setup() {

  //defining input pins for reading sensor values
  pinMode(firstSensorPin, INPUT);
  pinMode(seconSensorPin, INPUT);
  pinMode(thirdSensorPin, INPUT);

  //serial initialization
  Serial.begin(115200);
  Serial.println("SHT20 start");
  //initialization of sht20 sensor
  sht20.initSHT20();
  delay(100);
  sht20.checkSHT20();

  //used for LoRa communication
  while (!Serial);
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  Serial.println("LoRa Initial OK!");
}
```

Slika 3.2. – setup() funkcija programa koji služi za slanje (engl. transmitter)

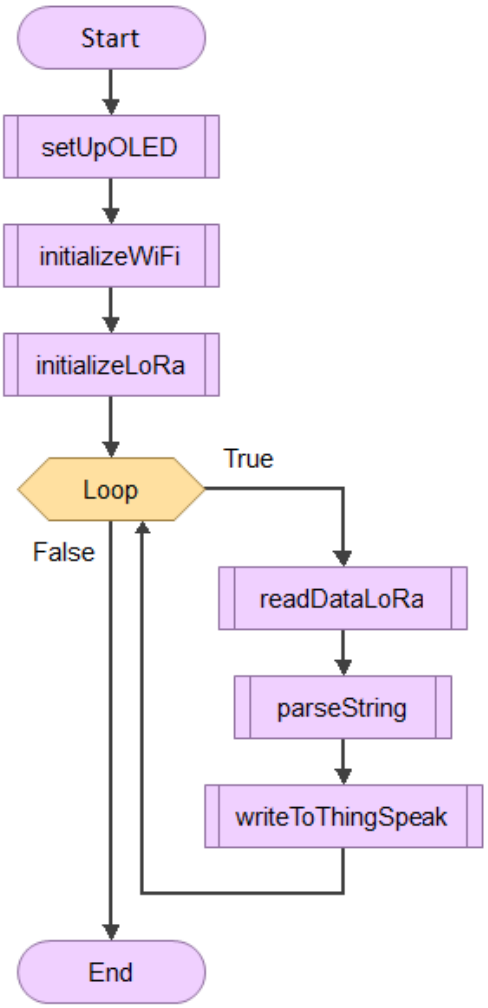
U loop() funkciji pozivaju se tri funkcije. Prva je readDataFromSensors(), koja čita vrijednosti sa sva 4 senzora te ih sprema u varijable. Iduća je mapValuesToPercentages() koja uzima vrijednosti senzora spremljenih u varijable te ih mapira iz brojčanih vrijednosti u postotke. Vrijednosti postotaka se kasnije šalju korištenjem LoRa-e na drugi ESP32 modul kako bi se dalje obradili i vizualizirali. Posljednja funkcija koja se poziva jest sendDataWithLoRa() koja uzima podatke izražene u postotcima te ih šalje korištenjem LoRa-e na drugi ESP32 upravljač.

Primjer koda za slanje podataka korištenjem LoRa-e vidljiv je na slici 3.3. Ovdje se kreira poruka koja se šalje na drugi ESP32 modul.

```
LoRa.beginPacket();
LoRa.print("|");
LoRa.print(percentageValueFirstSensor);
LoRa.print("|");
LoRa.print(percentageValueSecondSensor);
LoRa.print("|");
LoRa.print(percentageValueThirdSensor);
LoRa.print("|");
LoRa.print(humd);
LoRa.print("|");
LoRa.print(temp);
LoRa.print("| counter: ");
LoRa.print(counter);
LoRa.endPacket();
```

Slika 3.3. – prikaz slanja podataka korištenjem LoRa biblioteke

Osnovni model odnosno tok stanja programa koji se koristi za primanje (engl. *receiver*) može se vidjeti na slici 3.4.



Slika 3.4. - Tok dijagram programa za primanje podataka (engl. receiver)

Sa slike 3.4 je vidljivo da je kod koji se koristi za primanje organiziran u dva dijela, baš kao i kod koji se koristi za slanje. Pošto se također radi o Arduino programskom kodu i ovaj kod podijeljen je u dva glavna dijela. To su setup() i loop() funkcije. Setup() funkcija koristi se za inicijalno pokretanje funkcija i naredbi koje se moraju izvršiti jednom, prilikom pokretanja programa. Pošto se ESP32 upravljač koji se koristi za primanje podataka sa prvog ESP32 upravljača koristi i za slanje na ThingSpeak poslužitelj, mora se koristiti i WiFi modul za spajanje na lokalnu WiFi mrežu. SX1278 ESP32 upravljači koji se koriste za izradu ovog rada imaju na sebi integrirane OLED ekrane. Na ESP32 urpavljaču koji se koristi za primanje, podaci koji se primaju prikazuju se na OLED ekranu zbog lakše vizualizacije i otklanjanja pogrešaka.

Kako ESP32 upravljač koji se koristi za primanje podataka, te podatke prima putem LoRa-e, prikazuje ih na OLED ekranu, te ih šalje na ThinSpeak poslužitelj putem WiFi-ja, potrebno je inicijalizirati sve potrebne biblioteke i hardver koji se koristi za komunikaciju. Tako u setup() funkciji postoje tri funkcije koje se pozivaju jednom. To su setUpOLED() koja obavlja inicijalno postavljanje OLED ekrana, zatim initializeWiFi() koja obavlja inicijalno postavljanje WiFi modula, te initializeLoRa() koja obavlja inicijalno postavljanje LoRa modula. Na slici 3.5 vidljiv je način korištenja WIFi.h biblioteke za povezivanje na lokalnu WiFi mrežu.

```
//WiFi setup
WiFi.begin(ssid, password);

//WiFi connection
while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
```

Slika 3.5. – prikaz korištenja WiFi.h biblioteke za povezivanje na lokalnu WiFi mrežu

U loop() funkciji pozivaju se 3 funkcije; prva koja čita podatke koji stižu putem LoRa-e – readDataLoRa(), zatim parseString() koja obavlja parsiranje niza stringova koji dolaze putem LoRa-e, te ih dijeli u manje dijelove kako bi ih naposlijetku funkcija writeToThingSpeak() poslala na ThingSpeak poslužitelj. Prikaz loop() funkcije vidljiv je na slici 3.6.

```
//main function -- loops forever
void loop() {

  readDataLoRa();
  parseString(packet);
  writeToThingSpeak();
}
```

Slika 3.6. – prikaz loop() funkcije programa koji se koristi za primanje podataka (engl. receiver)

3.1. Problemi sa slanjem na ThingSpeak poslužitelj

Ono što je predstavljalo problem u slanju podataka na ThingSpeak jest metoda iz biblioteke koja se koristi za slanje podataka na poslužitelj. Na ThingSpeak poslužitelju postoje kanali (engl. *channels*) koji predstavljaju dijagrame u koje se upisuju vrijednosti sa senzora. U ovome radu korištena su 4 kanala za 4 senzora. ThingSpeak ima metodu writeField() koja uz predane vrijednosti za ID kanala, broj kanala, vrijednost koja se šalje te vrijednost API ključa, omogućava slanje podataka na kanal. Primjer iz koda vidljiv je na slici 3.7.

```
// Write value to Field 1 of a ThingSpeak Channel
int httpCode1 = ThingSpeak.writeField(myChannelNumber, 1, stringForUpload1, myWriteAPIKey);

if (httpCode1 == 200) {
  Serial.println("Channel write successful.");
}
else {
  Serial.println("Problem writing to channel. HTTP error code " + String(httpCode1));
}
```

Slika 3.7. – prikaz korištenja ThingSpeak biblioteke za slanje podataka na ThingSpeak poslužitelj

Problem je nastao tako što su se podaci uspješno slali na poslužitelj jednom, no poslužitelj nije slao potvrdu o uspješnosti slanja 200 OK kako bi se kod mogao dalje normalno izvršavati. Tada bi se program zaustavio i dalje nije bilo slanja podataka. Ono što je rezultat ovakvog koda jest da je moguće poslati samo jednu vrijednost sa senzora i to samo prvi puta kada se kod izvršava.

Rješenje ovog problema u načelu je vrlo jednostavno. Trebalo je napraviti HTTP POST zahtjev na lokalnom WiFi klijentu te ga uzastopno ponavljati sa novim vrijednostima. Isječak koda koji to obavlja nalazi se na slici 3.8.

```
dataToSend += "&field1=";
dataToSend += String(stringForUpload1);
dataToSend += "&field2=";
dataToSend += String(stringForUpload2);
dataToSend += "&field3=";
dataToSend += String(stringForUpload3);
dataToSend += "&field4=";
dataToSend += String(stringForUpload4);
dataToSend += "\r\n\r\n";

//posting data as a HTTP POST request
client.print("POST /update HTTP/1.1\n");
client.print("Host: api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(dataToSend.length());
client.print("\n\n");
client.print(dataToSend);
Serial.println("Data sent.");
```

Slika 3.8. – prikaz formiranja HTTP zahtjeva kojim se šalju podaci na ThingSpeak poslužitelj

Isti rezultat može se dobiti i kada bi se u bilo kojem Internet pregledniku upisala vezu oblika vidljivog na slici 3.9.

```
https://api.thingspeak.com/update.json?api_key=F8V6VSOLV80LLZV3&field1=12&field2=23%field3=54&field4=43
```

Slika 3.9. – prikaz veze (engl. link) kojim se šalju podaci na ThingSpeak poslužitelj

4. ANALIZA REZULTATA

Osnovni zahtjev rada bio je testirati nekoliko senzora za mjerenje vlažnosti na nekoliko različitih uzoraka zemlje. Prikupljena su tri uzorka zemlje; iz okolice Vinkovaca, sa Malog Lošinja te iz okolice Krapinskih Toplica. Na ovaj način prikupljeni su različiti tipovi zemlje, što pridonosi raznolikosti uzoraka na kojima se obavlja testiranje.

Kada se govori o tipovima zemlje, u Republici Hrvatskoj postoji pozamašan broj različitih tipova zemlje. Zemlja iz okolice Vinkovaca jest crnica, zemlja sa Malog Lošinja je crvenica, dok je tip zemlje iz okolice Krapinskih Toplica kambisol, odnosno gajnjača [23].

Crnicu karakterizira visok udio humusa, što ju čini iznimno povoljnom za sadnju i uzgoj raznih vrsta biljaka i žitarica. Crne je boje, odakle je i dobila naziv. Karakteristična je za panonske, sjeveroistočne dijelove Hrvatske, točnije Slavoniju. Crvenicu također karakterizira relativno visok udio humusa, što ju čini iznimno plodnom zemljom. Oksidi željeza i aluminija daju joj karakterističnu crvenu boju odakle je i dobila ime. Česta je u priobalnom području te jadranskim otocima. Kambisol ili gajnjača, karakteristični je tip zemlje koji se često pronalazi na brežuljkastim područjima u umjerenom podneblju, u Hrvatskoj do 400m visine. Uz crnicu i crvenicu spada u relativno plodna tla. Kambisol se u Hrvatskoj može pronaći na sjeverozapadu i u središnjoj Hrvatskoj.

Način na koji zemlja pripremljena za testiranje objašnjen je ukratko u idućem odlomku. Kako bi osigurani uvjeti bili što bliži idealnim (0% vlažnosti zemlje), zemlja koja je prikupljena za namjenu testiranja, prvotno je morala biti osušena. Način na koji je to izvedeno jest da je zemlja stavljena u pećnicu na 105°C na otprilike 2 sata [24]. To je osiguralo isparavanje velikog dijela vlažnosti iz zemlje. U idealnim uvjetima prije provođenja testiranja, vlažnost zemlje trebala bi biti što bliža 0%. Način na koji se može utvrditi referentna vrijednost vlažnosti zemlje objašnjena je u [24]. Ovaj postupak obavljen je za sva tri tipa zemlje. Prije sušenja zemlje u pećnici izmjerena je masa referentne zemlje koja je iznosila 400g. Uz masu posude od 375g, ukupna masa zemlje sa posudom iznosi 775g (vrijedi za sva tri tipa zemlje). Sva mjerenja mase izvedena su korištenjem kuhinjske vage preciznosti jednog grama.

Izračun vlažnosti zemlje je obavljen korištenjem formule 1:

$$M_C\% = \frac{W_2 - W_3}{W_3 - W_1} x100\%, \quad (3\text{-}1)$$

Gdje W_1 predstavlja masu posude u kojoj se nalazi zemlja, W_2 predstavlja masu zemlje prije sušenja zajedno sa masom posude, dok W_3 predstavlja masu zemlje nakon sušenja zajedno sa posudom.

Nakon sušenja masa zemlje zajedno sa posudom iznosila je 717g za crnicu, 731g za crvenicu te 723g za kambisol. Primjenom formule 1, dobiveni su rezultati vidljivi u tablici 4.1.

Tablica 4.1 – Izračun mase vlažnosti zemlje

Tip zemlje	Crnica	Crvenica	Kambisol
Masa prije sušenja (sa posudom)	775g	775g	775g
Masa poslije sušenja (sa posudom)	717g	731g	723g
Postotak vlažnosti	16,9%	12,36%	14,94%

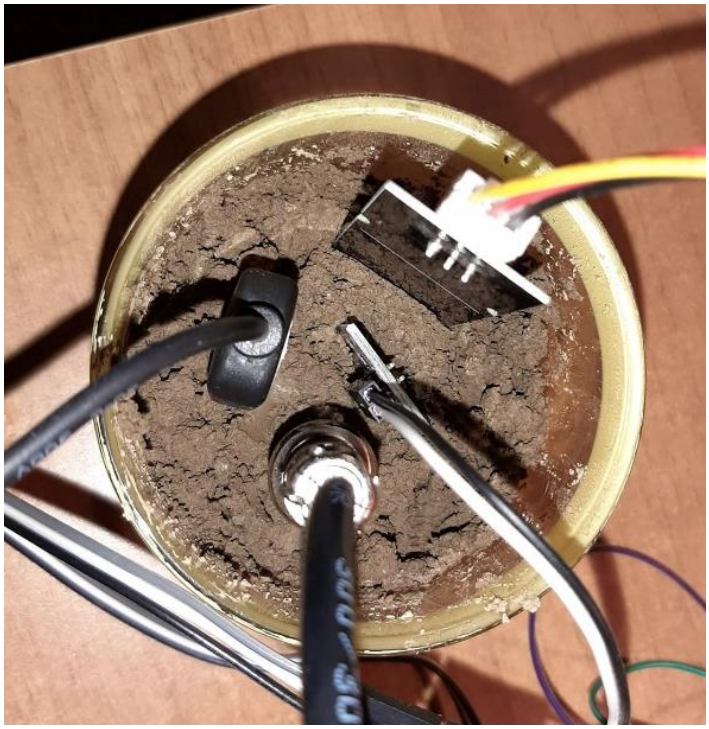
To znači da je vlažnost zemlje prije sušenja bila 16,9% za crnicu, 12,36% za crvenicu te 14,94% za kambisol. Nakon izračunavanja vlažnosti zemlje analitičkom metodom potrebno je isto učiniti i korištenjem senzorskog čvora. Testiranje će se obaviti na tri kontrolirane razine vlažnosti, a odabrane su: 15%, 25% i 35%.

Masa testnog uzorka zemlje korištenog prilikom izvođenja mjerenja korištenjem senzorskog čvora iznosi 160g. Prema tome, ukoliko je količina osušene zemlje 160g, postotak vlažnosti od 15% dobije se ulijevanjem 24g vode; analogno tome ulijevanjem 40g vode dobije se 25% vlažnosti, te ulijevanjem 56g vode dobije se vlažnost od 35%. Mjerenje mase vode vidljivo je na slici 4.1.

Slika 4.1. – Prikaz uzoraka vode

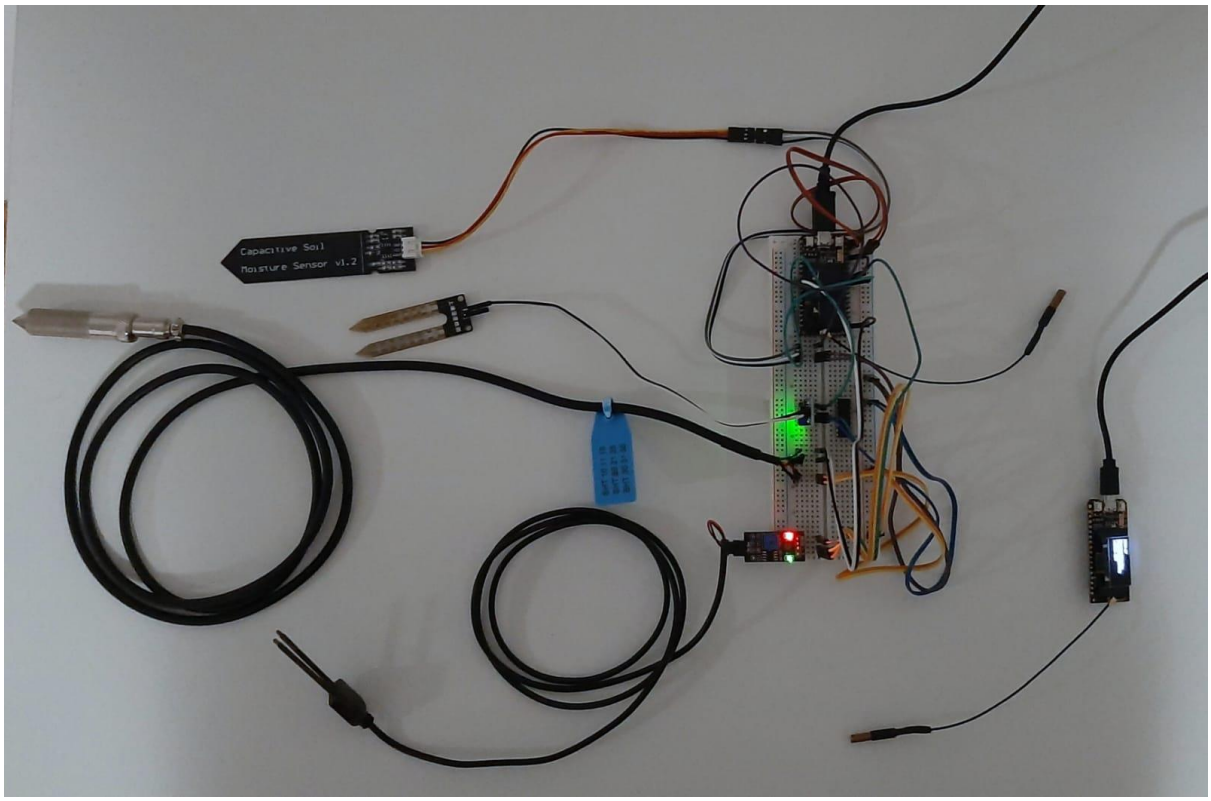
Rezultati mjerenja vlažnosti zemlje šalju se na ThingSpeak internet poslužitelj. Na ThingSpeak poslužitelju rezultati mjerenja se mogu prikazati na pre definiranim grafovima sa vremenskim oznakama mjerenja. Prilikom mjerenja vlažnosti zemlje potrebno je uzeti određenu količinu podataka te ju analizirati. Slobodnom procjenom, te promatranjem vrijednosti sa senzora, određeno je da je 30 uzoraka mjerenja sasvim dovoljno da se senzori stabiliziraju te da pokazuju dovoljno precizne i stabilne podatke, dok se s druge strane održava jednostavnost analiziranja podataka.

Prvi krug mjerenja obavljen je dodavanjem 24g vode u tri odvojene posude sa 160g suhe zemlje različitih vrsta. Drugi krug mjerenja obavljen je dodavanjem 40g vode u 160g suhe zemlje. Naposlijetku, treći krug mjerenja obavljen je dodavanjem 56g vode u 160g suhe zemlje. U svakoj od posuda nalazi se drugačiji tip zemlje; crnica, crvenica te kambisol. Nakon toga u svaku od posuda uneseni su senzori. Izgled testiranja gdje su sva 4 senzora umetnuta u zemlju vidljiv je na slici 4.2.



Slika 4.2. – Prikaz senzora uronjenih u zemlju prilikom testiranja

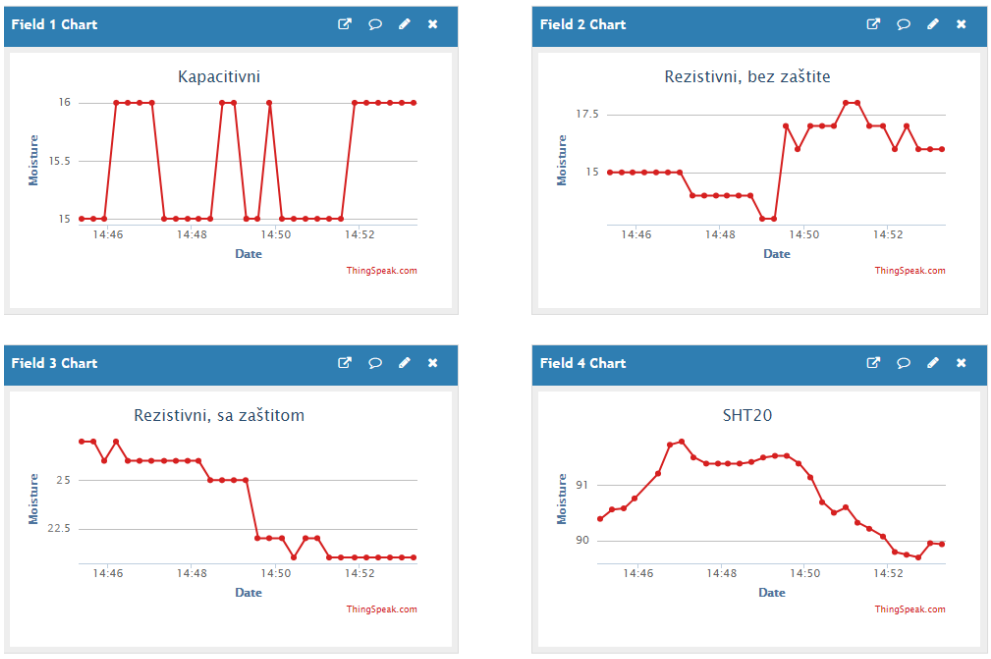
Na slici 4.3. vidljiv je izgled cijelog senzorskog čvora. Sastoji se od dva mikro upravljača te 4 senzora spojena na jedan od mikro upravljača.



Slika 4.3. – Stvarni izgled senzorskog čvora

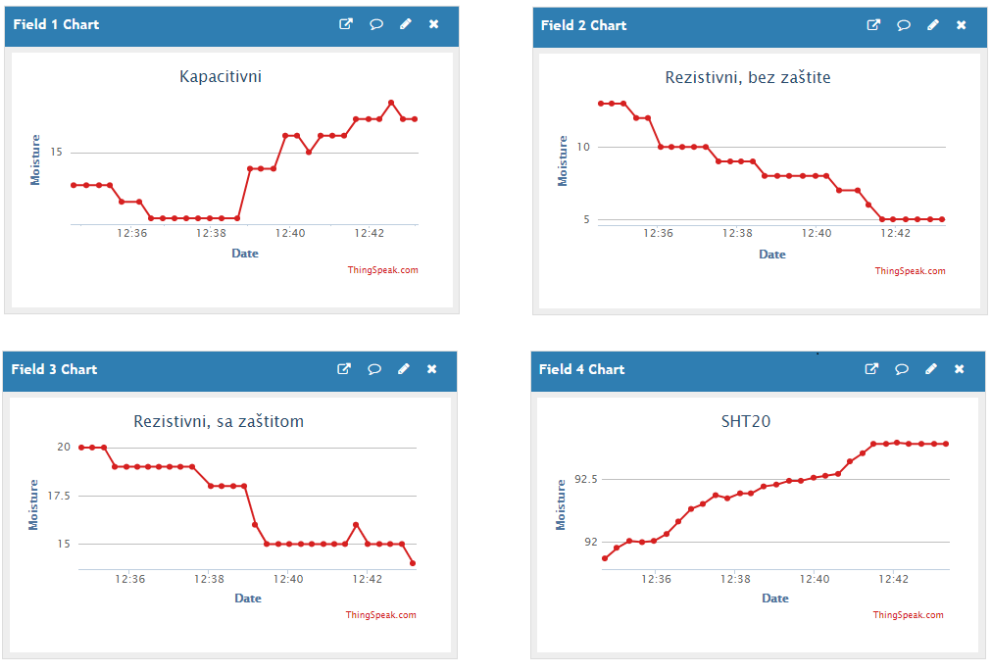
Zbog pojednostavljenja obrazloženja rezultata, izračunate su srednje vrijednosti pojedinog mjerenja. Mjerenje je izvršeno na tri vrste zemlje, sa četiri različita senzora, na tri različite referentne (izračunate) vlažnosti. Prvo će biti prikazani rezultati mjerenja za 15% vlažnosti, zatim 25% te naposlijetku 35%. Napomena: SHT20 senzor nije se uzimao u obzir prilikom računanja srednjih vrijednosti iz razloga što je pokazivao prevelike, odnosno netočne vrijednosti, pa bi ukupne srednje vrijednosti bile puno lošije zbog njega. SHT20 senzor je preosjetljiv za mjerenje vlažnosti u zemlji. On je prikladan isključivo za mjerenje vlažnosti zraka. SHT20 postaje preosjetljiv čim se uroni u zemlju koja ima u sebi određenu količinu vlažnosti, što ga je činilo neupotrebljivim za ovakvu vrstu mjerenja.

Prvo su prikazana mjerenja obavljena uz referentnu izračunatu vlažnost zemlje od 15%, za sve tri vrste zemlje, za sva 4 senzora. Na slici 4.4. nalaze se mjerenja za crnicu pri 15% vlažnosti. Za crnicu najtočnija mjerenja je imao rezistivni senzor, bez zaštite.



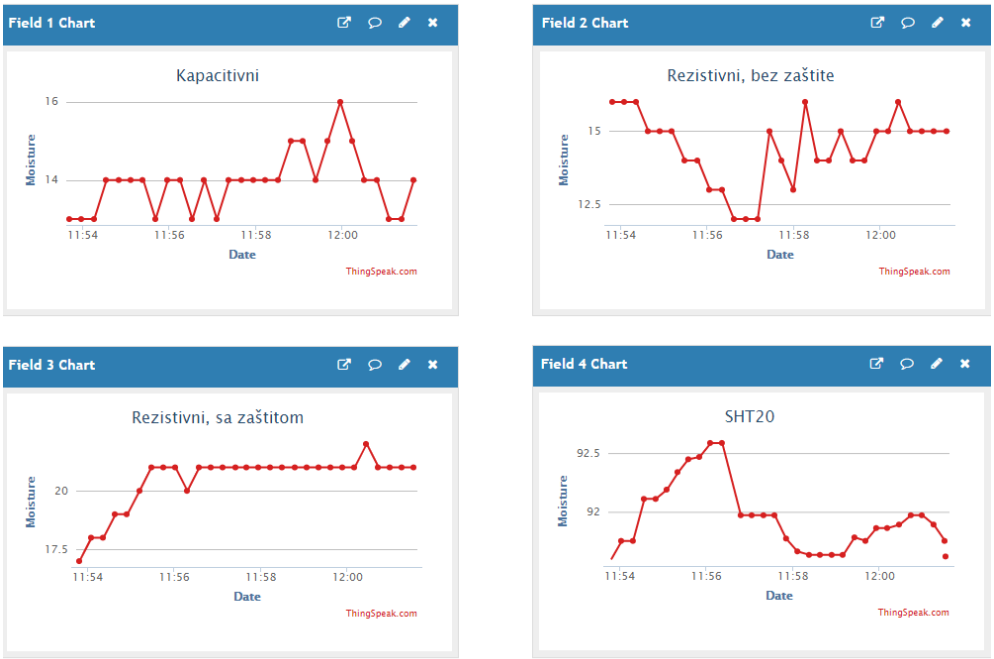
Slika 4.4. – Vrijednosti sa senzora za crnicu uz 15% referentnu vlažnost

Na slici 4.5. nalaze se mjerenja za crvenicu pri 15% vlažnosti. Za crvenicu najtočnija mjerenja je imao rezistivni senzor sa zaštitom.



Slika 4.5. – Vrijednosti sa senzora za crvenicu uz 15% referentnu vlažnost

Na slici 4.6. nalaze se mjerenja za kambisol pri 15% vlažnosti. Za kambisol najtočnije mjerenje je imao rezistivni senzor bez zaštite.



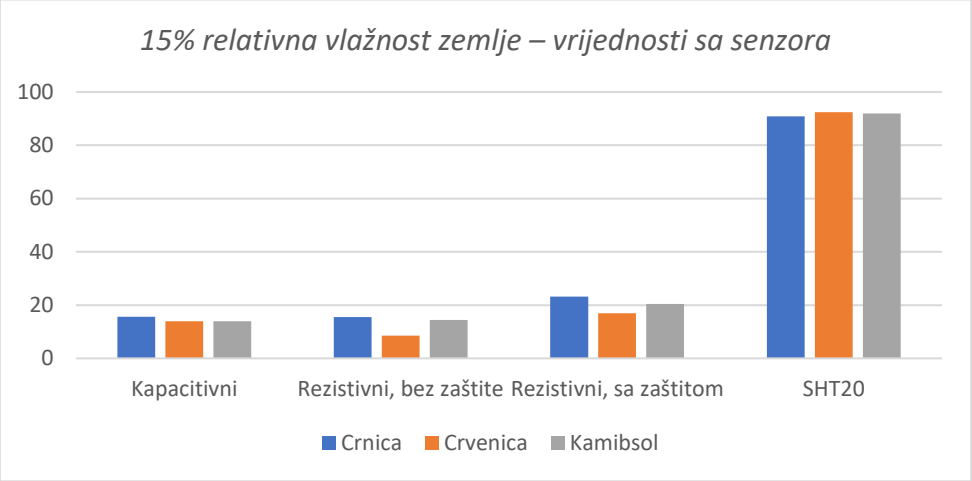
Slika 4.6. – Vrijednosti sa senzora za kambisol uz 15% referentnu vlažnost

Tablica 4.2. predstavlja srednje vrijednosti mjerenja obavljenih korištenjem sva 4 senzora za 15% referentnu vlažnost zemlje. Nema prevelikih odstupanja od ciljane vrijednosti od 15%.

Tablica 4.2. – 15% relativna vlažnost zemlje – vrijednosti sa senzora

Vrsta zemlje	Crnica	Crvenica	Kambisol
Kapacitivni	15,58%	13,93%	13,93%
Rezistivni, bez zaštite	15,52%	8,51%	14,41%
Rezistivni, sa zaštitom	23,24%	17%	20,48%
SHT20	90,81%	92,43%	91,86%

Dijagram 4.1. predstavlja grafički prikaz srednjih vrijednosti iz tablice 4.2. - mjerenja obavljenih korištenjem sva 4 senzora za 15% referentnu vlažnost zemlje.



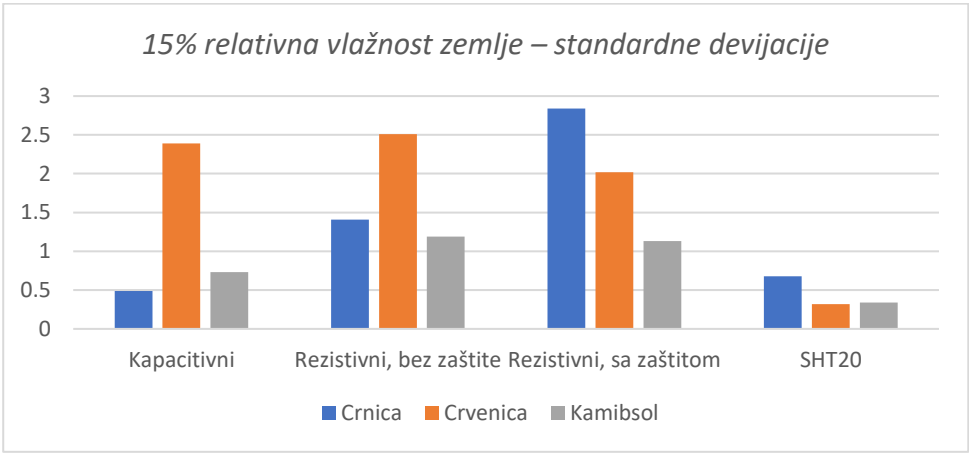
Dijagram 4.1. - 15% relativna vlažnost zemlje – vrijednosti sa senzora

Tablica 4.3. predstavlja standardne devijacije mjerenja sa sva 4 senzora za 15% referentnu vlažnost zemlje. Standardne devijacije iz tablice 4.3. pokazuju da su odstupanja od srednje vrijednosti u granicama normale, dok se malo povećanje vidi kod crvenice.

Tablica 4.3. – 15% relativna vlažnost zemlje - standardne devijacije

Vrsta zemlje	Crnica	Crvenica	Kambisol
Kapacitivni	0.49	2,39	0,73
Rezistivni, bez zaštite	1,41	2,51	1,19
Rezistivni, sa zaštitom	2,84	2,02	1,13
SHT20	0,68	0.32	0,34

Dijagram 4.2. predstavlja grafički prikaz standardnih devijacija iz tablice 4.3. - mjerenja obavljena korištenjem sva 4 senzora za 15% referentnu vlažnost zemlje.

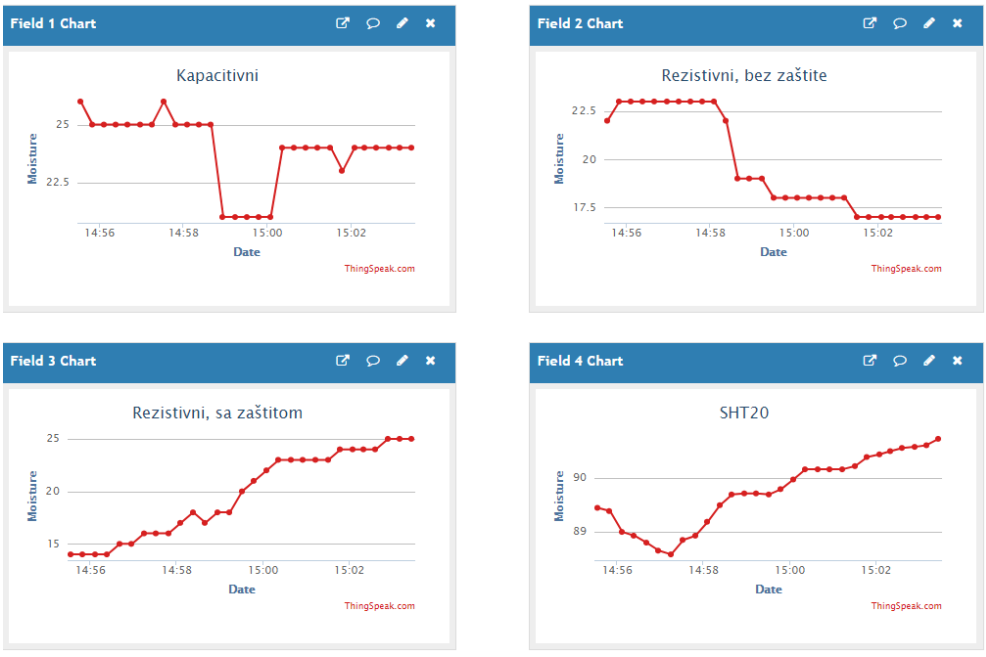


Dijagram 4.2. - 15% relativna vlažnost zemlje – standardne devijacije

Zatim su prikazana mjerenja obavljena uz referentnu izračunatu vlažnost zemlje od 25%, za sve tri vrste zemlje, za sva 4 senzora. Na slici 4.7. nalaze se mjerenja za crnicu pri 25% vlažnosti. Za crnicu najtočnija mjerenja je imao kapacitivni senzor.

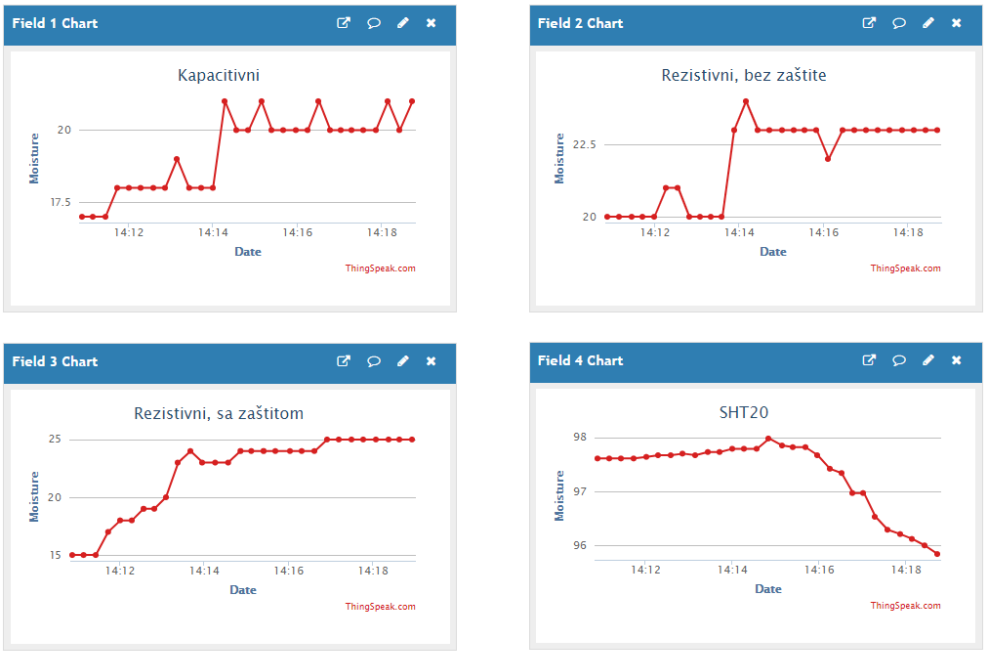
Slika 4.7. – Vrijednosti sa senzora za crnicu za 25% referentnu vlažnost

Na slici 4.8. nalaze se mjerenja za crvenicu pri 25% vlažnosti. Za crvenicu najtočnija mjerenja je imao kapacitivni senzor.



Slika 4.8. – Vrijednosti sa senzora za crvenicu za 25% referentnu vlažnost

Na slici 4.9. nalaze se mjerenja za kambisol pri 25% vlažnosti. Za kambisol najtočnije mjerenje je imao rezistivni senzor bez zaštite.



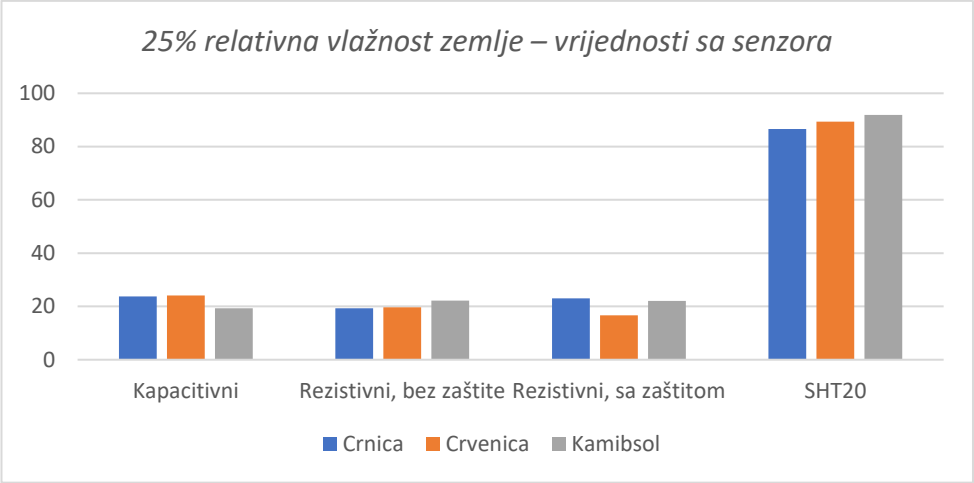
Slika 4.9. – Vrijednosti sa senzora za kambisol za 25% referentnu vlažnost

Tablica 4.4. predstavlja srednje vrijednosti mjerenja obavljenih korištenjem sva 4 senzora za 25% referentnu vlažnost zemlje. Vidljivo je da vrijednosti koje dolaze sa senzora najviše odstupaju od referente vrijednosti u odnosu na ostala mjerenja. Razlozi ovakvog ponašanja mogu biti razni. Od ljudske pogreške, do netočno izračunate referentne vrijednosti vlažnosti zemlje, pa sve do netočnog pokazivanja rezultata senzora, odnosno nepravilnog ponašanja senzora.

Tablica 4.4. – 25% relativna vlažnost zemlje - vrijednosti sa senzora

Vrsta zemlje	Crnica	Crvenica	Kambisol
Kapacitivni	23,75%	24,1%	19,28%
Rezistivni, bez zaštite	19,28%	19,65%	22,14%
Rezistivni, sa zaštitom	23%	19,68%	22,06%
SHT20	86,62%	89,41%	97,17%

Dijagram 4.3. predstavlja grafički prikaz srednjih vrijednosti iz tablice 4.4. - mjerenja obavljena korištenjem sva 4 senzora za 25% referentnu vlažnost zemlje.



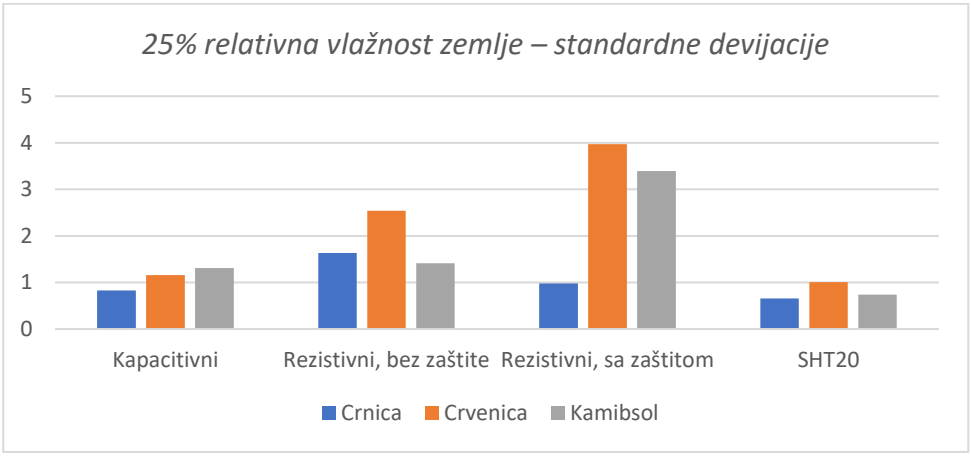
Dijagram 4.3. - 25% relativna vlažnost zemlje – vrijednosti sa senzora

Tablica 4.5. predstavlja standardne devijacije mjerenja sa sva 4 senzora za 25% referentnu vlažnost zemlje. Vidljivo je da su odstupanja od srednje vrijednosti u granicama normale.

Tablica 4.5. – 25% relativna vlažnost zemlje – standardne devijacije

Vrsta zemlje	Crnica	Crvenica	Kambisol
Kapacitivni	0,83	1,16	1,31
Rezistivni, bez zaštite	1,63	2,54	1,41
Rezistivni, sa zaštitom	0,98	3,97	3,39
SHT20	0,66	1,01	0,74

Dijagram 4.4. predstavlja grafički prikaz standardnih devijacija iz tablice 4.5. - mjerenja obavljena korištenjem sva 4 senzora za 25% referentnu vlažnost zemlje.

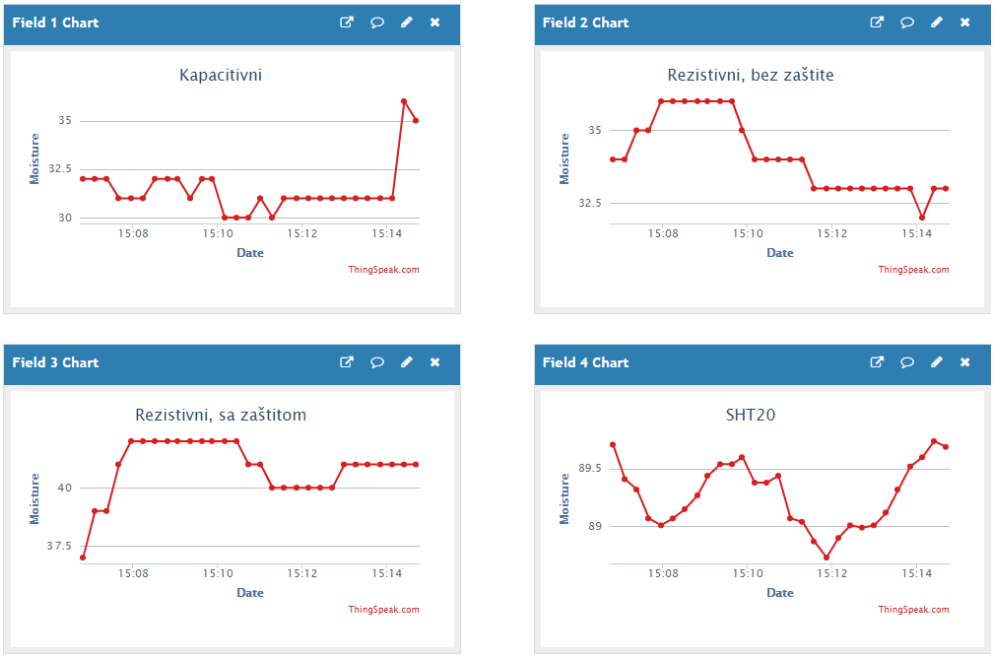


Dijagram 4.4. - 25% relativna vlažnost zemlje – standardne devijacije

Naposlijetku prikazana su mjerenja obavljena uz referentnu izračunatu vlažnost zemlje od 35%, za sve tri vrste zemlje, za sva 4 senzora. Na slici 4.10. nalaze se mjerenja za crnicu pri 35% vlažnosti. Za crnicu najtočnija mjerenja je imao kapacitivni senzor.

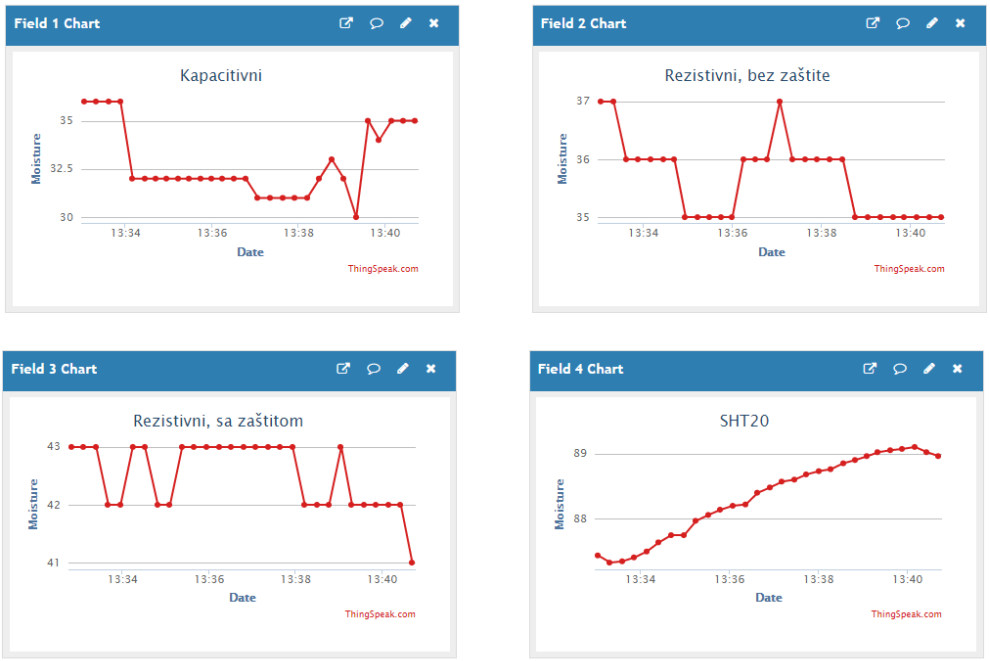
Slika 4.10. – Vrijednosti sa senzora za crnicu za 35% referentnu vlažnost

Na slici 4.11. nalaze se mjerenja za crvenicu pri 35% vlažnosti. Za crvenicu najtočnija mjerenja je imao rezistivni senzor bez zaštite.



Slika 4.11. – Vrijednosti sa senzora za crvenicu za 35% referentnu vlažnost

Na slici 4.12. nalaze se mjerenja za kambisol pri 35% vlažnosti. Za kambisol najtočnije mjerenje je imao rezistivni senzor bez zaštite.



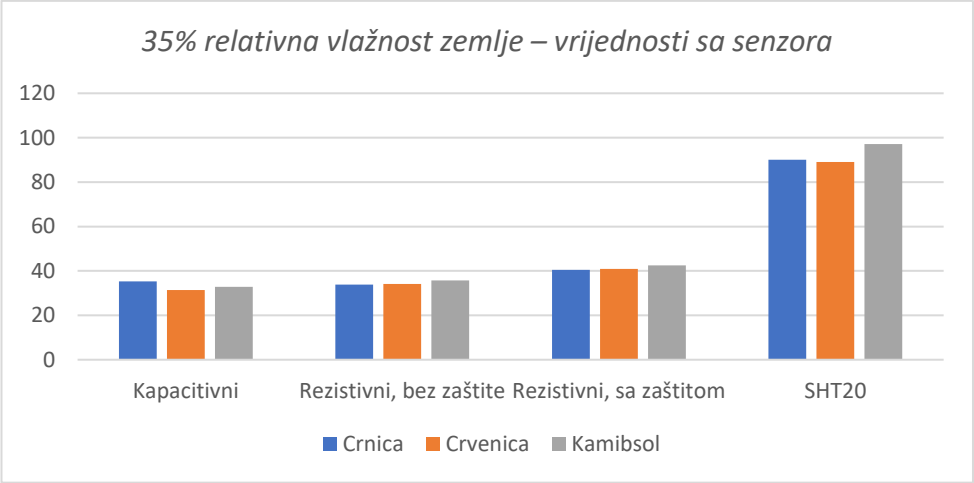
Slika 4.12. – Vrijednosti sa senzora za kambisol za 35% referentnu vlažnost

Tablica 4.6. predstavlja srednje vrijednosti mjerenja obavljenih korištenjem sva 4 senzora za 35% referentnu vlažnost zemlje. Iz tablice je vidljivo da su vrijednosti koje dolaze sa senzora unutar granica normale, odnosno dovoljno su blizu referente vrijednosti vlažnosti, da bi se moglo reći da je mjerenje obavljeno uz minimalna odstupanja.

Tablica 4.6. – 35% relativna vlažnost zemlje - vrijednosti sa senzora

Vrsta zemlje	Crnica	Crvenica	Kambisol
Kapacitivni	35,21%	31,44%	32,82%
Rezistivni, bez zaštite	33,89%	34,1%	35,65%
Rezistivni, sa zaštitom	40,41%	40,89%	42,52%
SHT20	90,1%	89.06%	87,93%

Dijagram 4.5. predstavlja grafički prikaz srednjih vrijednosti iz tablice 4.6. - mjerenja obavljena korištenjem sva 4 senzora za 35% referentnu vlažnost zemlje.



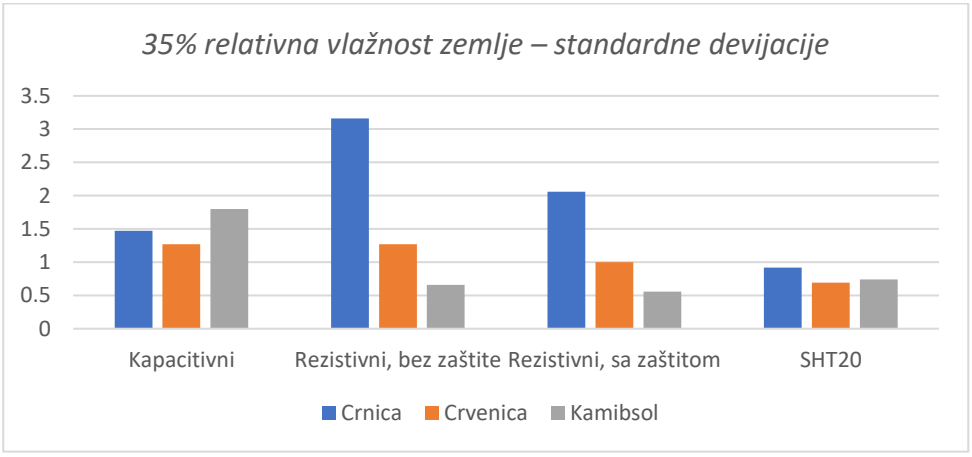
Dijagram 4.5. - 35% relativna vlažnost zemlje – vrijednosti sa senzora

Tablica 4.7. predstavlja standardne devijacije mjerenja sa sva 4 senzora za 35% referentnu vlažnost zemlje. Vrijednosti iz tablice 4.7. pokazuju da su odstupanja od srednje vrijednosti u granicama normale.

Tablica 4.7. – 35% relativna vlažnost zemlje – standardne devijacije

Vrsta zemlje	Crnica	Crvenica	Kambisol
Kapacitivni	1,47	1,27	1,8
Rezistivni, bez zaštite	3,16	1,27	0.66
Rezistivni, sa zaštitom	2,06	1,2	0,56
SHT20	0,92	0,69	0,72

Dijagram 4.6. predstavlja grafički prikaz standardnih devijacija iz tablice 4.7. - mjerenja obavljena korištenjem sva 4 senzora za 35% referentnu vlažnost zemlje.



Dijagram 4.6. - 35% relativna vlažnost zemlje – standardne devijacije

Prosječne vrijednosti vlažnosti svih senzora po vrsti zemlje za svaku razinu vlažnosti vidljive su u tablici 4.8.

Tablica 4.8. – prosječna mjerenja vlažnosti svih senzora zajedno za svaku vrstu zemlje u odnosu na izračunatu vlažnost

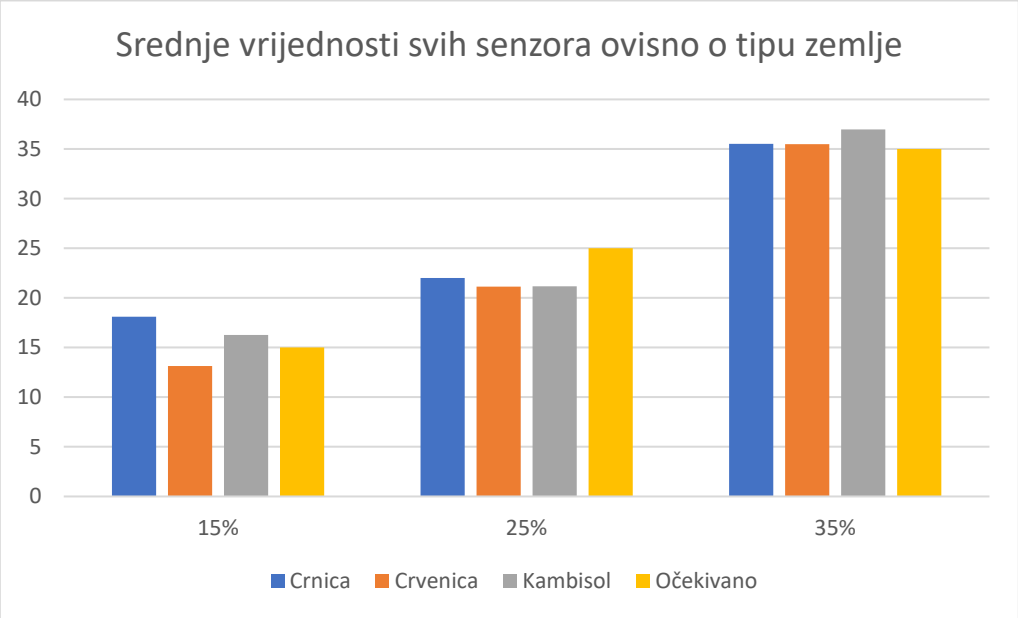
Vrsta zemlje	Crnica	Crvenica	Kambisol
15%	18,11%	13,15%	16,27%
25%	22,01%	21,14%	21,16%
35%	35,51%	35,48%	36,99%

Ukupna prosječna, srednja vrijednost vlažnosti sa senzora za predefinirane, referentne postotke vlažnosti dana je u tablici 4.9.

Tablica 4.9. – ukupna prosječna vrijednost izmjerene vlažnosti u odnosu na referentne vrijednosti vlažnosti

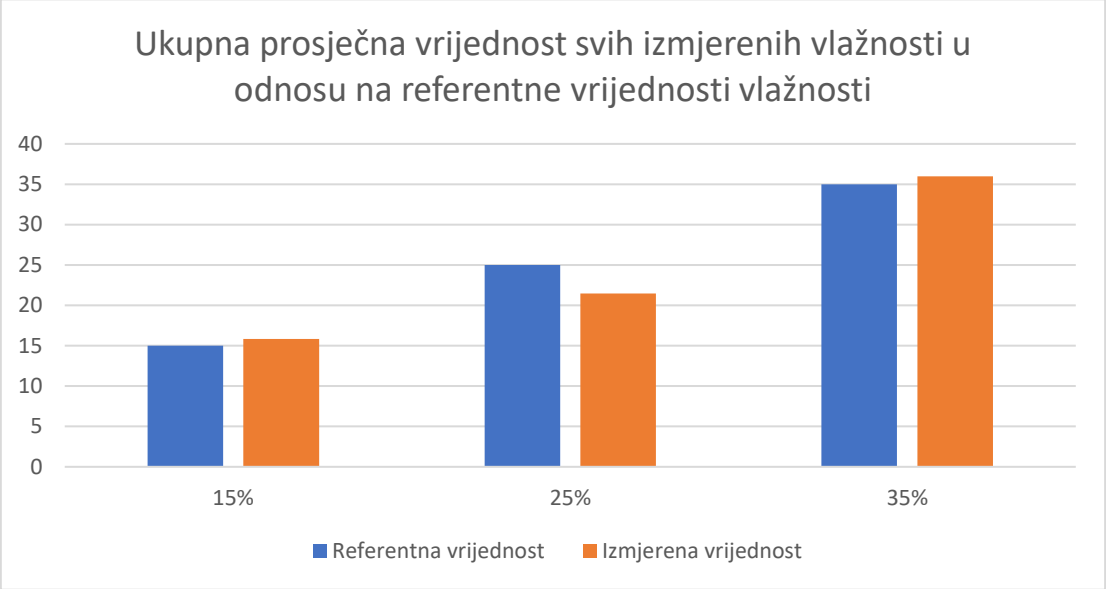
Referentna vrijednost	15%	25%	35%
Izmjerena vrijednost	15,84%	21,47%	35,99%

Na dijagramu 4.7., prema tablici 4.8. grafički su prikazane srednje vrijednosti sa svih senzora ovisno o tipu zemlje. Vidljivo je da su najtočniji rezultati uz 35% vlažnosti.



Dijagram 4.7. – Srednje vrijednosti svih senzora ovisno o tipu zemlje

Na dijagramu 4.8., prema tablici 4.9 prikazana je prosječna vrijednost izmjerenih vlažnosti u odnosu na referente, izračunate vrijednosti vlažnosti. Vidljivo je da je najveće odstupanje od izračunate vrijednosti najveće kod vlažnosti od 25%, dok je najmanje odstupanje kod vrijednosti od 15%.



Dijagram 4.8. - Ukupna prosječna vrijednost svih izmjerenih vlažnosti u odnosu na referentne vrijednosti vlažnosti

Sa gledišta svih mjerenja za sve vrste zemlje, za sve referentne vrijednosti vlažnosti, najtočnije rezultate pokazao je rezistivni senzor bez zaštite. U najvećem broju mjerenja pokazao je rezultate najbliže referentnima. Daleko najlošije rezultate pokazao je SHT20 koji se prilikom analize rezultata nije niti uzimao u obzir.

5. ZAKLJUČAK

Uzimajući u obzir poglavlje 3.2. gdje je objašnjena mana rezistivnih senzora, te uzimajući u obzir dobivene rezultate mjerenja sa senzora, može se doći do sljedećih zaključaka. Za povremena kratka mjerenja najtočniji je rezistivni senzor bez zaštite. No kako će ovakvi senzori nakon kratkog vremena početi pokazivati netočne rezultate mjerenja iz razloga objašnjenog u poglavlju 3.2., preporuka je za mjerenja vlažnosti zemlje koristiti kapacitivni senzor. Kod kapacitivnog senzora nema direktnog prolaska struje kroz zemlju, nego se mjeri kapacitet zemlje. Samim time nema opasnosti od trošenja, odnosno oksidacije senzorskih elektroda. SHT20 senzor pokazao se kao najlošiji za mjerenje vlažnosti zemlje. Taj senzor pokazao se kao preosjetljiv za mjerenje vlažnosti zemlje, unatoč robusnom i kvalitetnom kućištu. SHT20 je prije mjerenja, odnosno prije uranjanja u zemlju pokazivao vlažnost između 50% i 60%. Ta vlažnost odgovara vlažnosti zraka u prostoriji u kojoj su se provodila mjerenja. SHT20 primjeren je za mjerenje vlažnosti zraka. Nije uočena očita povezanost vrste zemlje i točnosti senzora. Međutim, na referentnom mjerenju, uočeno je da uzorak crnice ima najveću vlažnost, zatim slijedi kambisol te naposlijetku crvenica. Uzrok tome može biti da je nekoliko dana prije uzimanja uzoraka zemlje padala kiša, pa je zemlja ostala natopljena, što je uzrokovalo veću količinu vlažnosti u zemlji. Analogno tome, uzrok niže vlažnosti zemlje za kambisol i crvenicu može biti da duži period prije uzimanja uzorka nije padala kiša, pa je zemlja imala priliku osušiti se. Također, referentne vrijednosti će varirati ovisno o godišnjem dobu. Svi uzorci zemlje uzeti su između 1.7.2019. i 15.7.2019. Da su uzorci uzeti u jesen kada ima puno više padalina, i referente vrijednosti vlažnosti dobivene izračunom, bile bi ukoliko veće.

Naposlijetku, izrađeni senzorski čvor pogodan je za mjerenje i analiziranje vlažnosti tla na udaljenim i zabačenim mjestima, jer omogućava direktnu komunikaciju dva mikro upravljača korištenjem LoRa komunikacije. Kao najbolji od testiranih senzora za mjerenje vlažnosti tla, uzimajući u obzir poglavlje 3.2., pokazao se kapacitivni senzor, dok su rezistivni senzori dobri za povremeno korištenje isključivo u laboratorijskim uvjetima.

6. POPIS FORMULA

Formula (3-1) za izračunavanje postotka vlažnosti u uzorku zemlje.

$$M_C\% = \frac{W_2 - W_3}{W_{3-W_1}} x100\%, \quad (3\text{-}1)$$

7. LITERATURA

[1] https://www.eenewseurope.com/news/lora-connected-soil-moisture-sensors-help-farmers-cut-water-consumption# - Članak o LoRa tehnologiji i senzorima koji prate vlažnost zemlje, 20.08.2019.

[2] https://www.kerlink.com/ - Službena stranica Kerlink tvrtke, 20.08.2019.

[3] C. Li, C. Yang, Authenticated Key Management Protocols for Internet of Things, 2018.

[4] A. Lavric, V. Popa, Internet of Things and LoRa™ Low-Power WideArea Networks: A Survey

[5] O. Perešini, T. Krajkovič, More efficient IoT communication through LoRa network with LoRa@FIIT and STIOT protocols, 2017.

[6] V. A. Dambal, S. Mohadikar, A. Kumbhar, I. Guvenc, Improving LoRa Signal Coverage in Urban and Sub-Urban Environments with UAVs

[7] https://www.semtech.com/lora/what-is-lora - Službena stranica Semtech tvrtke koja se bavi proizvodnjom LoRa chipova, 12.08.2019.

[8] https://lora-alliance.org/ - Službena LoRa Alliance web stranica, 13.08.2019.

[9] A. Springer, W. Gugler, M. Huemer, L. Reindl, C.C.W. Rupper, R. Weigel, Spread Spectrum Communications Using Chirp Signals, 2000.

[10] https://www.youtube.com/watch?v=lg0eZWZFKiE - YouTube video sa objašnjenjem „chirp spread spectrum“ modulacije, 15.08.2019.

[11] https://www.thethingsnetwork.org/docs/lorawan/limitations.html - Ograničenja LoRa tehnologije, 16.08.2019.

[12] https://thingspeak.com/pages/learn_more - ThingSpeak web poslužitelj na kojem se podaci sa senzora šalju i obrađuju, 27.08.2019.

[13] https://www.banggood.com/2Pcs-LILYGO-TTGO-LORA-SX1278-ESP32-0_96-OLED-Display-Module-16-Mt-Bytes-128-Mt-bit-433Mhz-For-Arduino-p-1270420.html?rmmds=detail-left-hotproducts__5&cur_warehouse=CN - Web stranica sa koje je slinuta slika ESP32, 28.08.2019.

[14] https://www.hackster.io/devashish-gupta/capacitive-v-s-resistive-soil-moisture-sensor-e241f2 - Usporedba kapacitivnih i rezistivnih senzora, 29.08.2019.

[15] https://www.makerfabs.com/soil-moisture-sensor-module.html - Web stranica sa koje je skinuta slika senzora, 30.08.2019.

[16] https://www.makerfabs.com/soil-moisture-sensor.html - Web stranica sa koje je skinuta slika senzora, 30.08.2019.

[17] https://www.makerfabs.com/soil-humidity-sensor-capacitive.html - Web stranica sa koje je skinuta slika senzora,30.08.2019.
[18] https://www.youtube.com/watch?v=udmJyncDvw0 - YouTube video iz kojega je uzeta slika sheme senzora, 30.08.2019.
[19] https://www.makerfabs.com/soil-temperature-and-humidity-sensor-sht20.html - Web stranica sa koje je skinuta slika senzora, 30.08.2019.
[20] D. Bonači, I^2C Sabirnica, 2007.
[21] https://raspberrypi.stackexchange.com/questions/68133/is-soil-moisture-sensor-corrosion-normal - Objašnjenje oksidacije rezistivnih senzora, 05.09.2019.
[22] https://github.com/hrvojekok/soilMoistureMonitoring - Github stranica na kojoj se može pronaći kompletan kod rada, 30.08.2019.
[23]https://bpb.opendns.com/b/https/hr.metapedia.org/wiki/Hrvatska_tla_(zemlji%C5%A1te) - Opis vrste zemlje u Hrvatskoj, 05.09.2019.
[24] Department of Sustainable Natural Resources, Soil Survey Standard Test Method

8. SAŽETAK

Sve bržim razvojem tehnologije raste i njezin utjecaj na okoliš. Napredak tehnologije može se između ostaloga, usmjeriti ka povećanju brige za okoliš. Jedan od načina kako se briga za okoliš može povećati jest promatranje pojava i stanja u samom okolišu, te djelovanje sukladno tim pojavama. Internet objekata (engl. *Internet of Things*) omogućuje razvijanje sustava za nadzor i upravljanje različitim senzorima i aktuatorima iz ljudske okoline. Jedan od glavnih aspekata ljudske brige za okoliš jest i briga o vlažnosti zemlje. Vlažnost zemlje može biti pokazatelj koliko je neko tlo pogodno za sadnju i uzgoj biljaka, odnosno koliko je neko tlo pogodno za izgradnju građevina. U sklopu rada razvijen je senzorski čvor koji omogućuje mjerenje, kontrolu, nadzor i vizualizaciju podataka sa senzora za mjerenje vlažnosti tla. Pružen je uvid u nekoliko vrsta zemlje, te su uspoređena 4 različita senzora. Na kraju su uspoređeni senzori za mjerenje vlažnosti tla te je najbolje rezultate dao rezistivni senzor bez zaštite. No, uzimajući u obzir njegovu dugotrajnost, kao najprikladniji senzor za mjerenje vlažnosti tla odabran je kapacitivni senzor. On omogućava relativno točne rezultate mjerenja uz dovoljno visoku razinu pouzdanosti i izdržljivosti.

Ključne riječi: Internet of Things, C++, ESP32 SX1278, senzori, vlažnost, okoliš, tlo, LoRa, WiFi

9. NASLOV I SAŽETAK NA ENGLESKOM JEZIKU

DEVELOPMENT AND TESTING OF A SENSOR NODE FOR SOIL MOISTURE ANALYSIS

With the rapid development of technology, its impact on the environment is also growing. The advancement of technology can, among other things, be directed towards increasing environmental concerns. One way to increase environmental concerns is to observe the phenomena and conditions in the environment itself, and to act on those phenomena. The Internet of Things enables the development of systems for monitoring and controll of various sensors and actuators from the environment. One of the major aspects of concern for the environment is the care of soil moisture. The moisture of the soil can tell how much that particular soil is suitable for planting and growing plants, or how much that soil is suitable for building real estate. As part of the thesis, a sensor node was developed that enables measurement, control, monitoring and visualization of data from soil moisture sensors. Several soil types were provided and 4 different sensors were compared. Finally, the soil moisture sensors were compared and the best results were obtained by a resistive sensor without the protective layer around it. But considering its longevity, a capacitive sensor was chosen as the most suitable sensor for measuring soil moisture. It provides relatively accurate measurement results with a sufficiently high level of reliability and durability.

10. ŽIVOTOPIS

Hrvoje Kokošarević rođen je 2.9.1995. u Starim Mikanovcima. 2009. godine završava Osnovnu školu Vladimira Nazora u Vinkovcima. Gimnaziju Matije Antuna Reljkovića u Vinkovcima upisuje 2009., a završava 2014. Iste godine upisuje, a 2017. godine završava preddiplomski studij Elektrotehnike, smjer Komunikacije i informatika, na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. 2017. godine upisuje sveučilišni diplomski studij Elektrotehnike, smjer Mrežne tehnologije.

Hrvoje Kokošarević

11. PRILOZI

U nastavku se nalazi programski kod koji služi za slanje podataka sa senzora - *sender*.

```
#include <SPI.h>
#include <LoRa.h>
#include<Arduino.h>
#include "SSD1306.h"
#include <Wire.h>
#include "DFRobot_SHT20.h"

//defining pins for LoRa module initialization
#define SCK     5    // GPIO5  -- SX1278's SCK
#define MISO    19   // GPIO19 -- SX1278's MISO
#define MOSI    27   // GPIO27 -- SX1278's MOSI
#define SS      18   // GPIO18 -- SX1278's CS
#define RST     14   // GPIO14 -- SX1278's RESET
#define DI0     26   // GPIO26 -- SX1278's IRQ(Interrupt Request)
#define BAND    433E6  //433MHz frequency for LoRa in EU

//defining pins for sensors
int firstSensorPin = 36;
int secondSensorPin = 35;
int thirdSensorPin = 34;

//counter counts number of packets sent -- it is displayed on the OLED
int counter = 0;

//these variables are used for reading sensor values to
int firstSensorValue = 0;
int secondSensorValue = 0;
int thirdSensorValue = 0;

//initialization of sht20 library object
DFRobot_SHT20 sht20;

void setup() {

  //defining input pins for reading sensor values
```

```
  pinMode(firstSensorPin, INPUT);
  pinMode(seconSensorPin, INPUT);
  pinMode(thirdSensorPin, INPUT);

  //serial initialization
  Serial.begin(115200);
  Serial.println("SHT20 start");
  //initialization of sht20 sensor
  sht20.initSHT20();
  delay(100);
  sht20.checkSHT20();

  //used for LoRa communication
  while (!Serial);
  SPI.begin(SCK, MISO, MOSI, SS); //when compiled - gives an error, but it
uploads without any problems
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  Serial.println("LoRa Initial OK!");
}


//main function -- repeats forever
void loop() {

  //reading values from sht20 sensor -- using sht20 library methods
  float humd = sht20.readHumidity();
  float temp = sht20.readTemperature();
  Serial.print("Time:");
  Serial.print(millis());
  Serial.print(" Temperature:");
  Serial.print(temp, 1);
  Serial.print("C");
  Serial.print(" Humidity:");
  Serial.print(humd, 1);
  Serial.print("%");
  Serial.println();
```

```
  //reading values from sensors with analogRead
  firstSensorValue = analogRead(firstSensorPin);
  secondSensorValue = analogRead(secondSensorPin);
  thirdSensorValue = analogRead(thirdSensorPin);
  //Serial.println(firstSensorValue);
  //Serial.println(secondSensorValue);
  //Serial.println(thirdSensorValue);

  //when using capacitive soil moisture sensor use
convertToPercentagesCapacitive function
  //sensors read analog values, but input pins receive values from 0 to
4095,
  //therefore those values need to be converted to percetages to be
displayes properly
  int percentageValueFirstSensor =
convertToPercentagesCapacitive(firstSensorValue);
  int percentageValueSecondSensor =
convertToPercentages(secondSensorValue);
  int percentageValueThirdSensor = convertToPercentages(thirdSensorValue);

  Serial.println(percentageValueFirstSensor);
  Serial.println(percentageValueSecondSensor);
  Serial.println(percentageValueThirdSensor);

  Serial.print("Sending packet: ");
  //Serial.println(counter);

  //sending data via LoRa
  //data that is sent via LoRa is only percentage values
  LoRa.beginPacket();
  LoRa.print("|");
  LoRa.print(percentageValueFirstSensor);
  LoRa.print("|");
  LoRa.print(percentageValueSecondSensor);
  LoRa.print("|");
  LoRa.print(percentageValueThirdSensor);
  LoRa.print("|");
  LoRa.print(humd);
  LoRa.print("|");
  LoRa.print(temp);
  LoRa.print("| counter: ");
```

```
  LoRa.print(counter);
  LoRa.endPacket();
  delay(300);
  counter++;
  firstSensorValue = 0;
  secondSensorValue = 0;
  thirdSensorValue = 0;

}


//all sensors should use convertToPercentages function
//capacitive sensor never reaches maximum value of 4095 even though it
should
//capacitive sensor uses convertToPercentagesCapacitive

//when using capacitive soil moisture sensor use
convertToPercentagesCapacitive function
int convertToPercentagesCapacitive(int value){

  int percentage;
  percentage = map(value, 0, 4095, 100, 10); //used to be 3450 -- not good
  return percentage;

}

//when using resistive sensors use convertToPercentages function
int convertToPercentages(int value) {

  int percentage;
  percentage = map(value, 4095, 0, 10, 100);
  return percentage;

}
```

U nastavku se nalazi programski kod koji služi za primanje podataka – *receiver*.

```
#include <SPI.h>
```

```
#include <LoRa.h>
#include <Wire.h>
#include "SSD1306.h"
#include <WiFi.h>
#include "ThingSpeak.h"
#include "secrets.h"


//wifi password and ssid
//const char* ssid     = "6486EF";
//const char* password = "EVW32C0S00021595";
//const char* ssid     = "AndroidHrvoje";
//const char* password = "hrvojekokkok";
const char* ssid     = "TRESNJEVACKI_MALISANI";
const char* password = "subidubidubi";

//thingspeak web page link
const char* host = "api.thingspeak.com";

//API key provided from thingspeak
String apiKey = "F8V6VSOLV80LLZV3";

//used for setting up the SX1278 LoRa module
#define SCK     5    // GPIO5  -- SX1278's SCK
#define MISO    19   // GPIO19 -- SX1278's MISO
#define MOSI    27   // GPIO27 -- SX1278's MOSI
#define SS      18   // GPIO18 -- SX1278's CS
#define RST     14   // GPIO14 -- SX1278's RESET
#define DI0     26   // GPIO26 -- SX1278's IRQ(Interrupt Request)
#define BAND    433E6 //915E6 - frequency of LoRa 433MHz/915MHz

//network key index number (needed only for WEP)
int keyIndex = 0;
WiFiClient client;
```

```
//setting up the OLED dipslay
SSD1306 display(0x3c, 4, 15);

//defines signal strength RSSI -- received signal strength indicator
String rssi = "RSSI --";
String packSize = "--";
String packet;

//these are the strings in wich values from sensors are parsed
//these values are uploaded to ThingSpeak
String stringForUpload1;
String stringForUpload2;
String stringForUpload3;
String stringForUpload4;
String stringForUpload5;

//edits values for displaying on OLED display
void loraData(){

  display.clear();
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.setFont(ArialMT_Plain_10);
  display.drawString(0 , 15 , "Received "+ packSize + " bytes"); //remove
after adding all sensors
  display.drawStringMaxWidth(0 , 26 , 128, packet);
  display.drawString(0, 0, rssi);
  display.display();
}




void setup() {

  Serial.begin(9600);
```

```
  //OLED display setup
  pinMode(16,OUTPUT);
  digitalWrite(16, LOW);    // set GPIO16 low to reset OLED
  delay(50);
  digitalWrite(16, HIGH); // while OLED is running, must set GPIO16 in high
`

  //WiFi setup
  WiFi.begin(ssid, password);

  //WiFi connection
  while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  WiFi.mode(WIFI_STA);

  //initializes thingspeak client
  ThingSpeak.begin(client);

  //LoRa initialization
  while (!Serial);
  Serial.println();
  Serial.println("LoRa Receiver Callback");
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
```

```
  //LoRa.onReceive(cbk);
  LoRa.receive();
  Serial.println("init ok");
  display.init();
  display.flipScreenVertically();
  display.setFont(ArialMT_Plain_10);

  delay(1500);

}

//this function sends data to ThingSpeak server
void writeToThingSpeak(){

  Serial.println("Prepare to send data");

  //use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;

  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;

    } else {
    //fields are referred to thingspeak server diagrams
    String dataToSend = apiKey;
    dataToSend += "&field1=";
    dataToSend += String(stringForUpload1);
    dataToSend += "&field2=";
    dataToSend += String(stringForUpload2);
    dataToSend += "&field3=";
    dataToSend += String(stringForUpload3);
```

```
    dataToSend += "&field4=";
    dataToSend += String(stringForUpload4);
    dataToSend += "\r\n\r\n";

    //posting data as a HTTP POST request
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(dataToSend.length());
    client.print("\n\n");
    client.print(dataToSend);
    Serial.println("Data sent.");

    delay(2000);
  }
  client.stop();

}


//used for reading data comming from other LoRa module device
void cbk(int packetSize) {

  packet ="";
  packSize = String(packetSize,DEC);
  for (int i = 0; i < packetSize; i++) {
    packet += (char) LoRa.read();
  }
  rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
  loraData();
```

```
}

//used for reading data comming from other LoRa module device
void readDataLoRa(){

  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    cbk(packetSize);
    }
  delay(10);
  Serial.println(packet);

}

//main function -- loops forever
void loop() {

  readDataLoRa();
  parseString(packet);
  writeToThingSpeak();

}

//this function parses packets comming from LoRa module
void parseString(String packet){

  //parsing strings comming from LoRa
  stringForUpload1 = packet.substring(1,3);
  stringForUpload2 = packet.substring(4,6);
  stringForUpload3 = packet.substring(7,9);
  stringForUpload4 = packet.substring(10,15);
  //not used -- isn't sent to thinspeak -- this is the temperature from
SHT20
  stringForUpload5 = packet.substring(16,21);
  Serial.println(stringForUpload1);
```

```
  Serial.println(stringForUpload2);
  Serial.println(stringForUpload3);
  Serial.println(stringForUpload4);
  //not used -- isn't sent to thinspeak -- this is the temperature from
SHT20
  Serial.println(stringForUpload5);


}
```

U nastavku se nalazi skripta secrets.h u koju su spremljeni podaci sa ThingSpeak poslužitelja.

```
// Use this file to store all of the private credentials
// and connection details


//#define SECRET_SSID "6486EF" /replace MySSID with your WiFi network name
//#define SECRET_PASS "EVW32C0S00021595" /replace MyPassword with your WiFi
password


#define SECRET_SSID "TRESNJEVACKI_MALISANI"
#define SECRET_PASS "subidubidubi"


//#define SECRET_SSID "Tech_D0040587"
//#define SECRET_PASS "CUMBBWRE"


#define SECRET_CH_ID // replace 0000000 with your channel number
#define SECRET_WRITE_APIKEY "RT1BRWVZ39O6DD35" //replace XYZ with your
channel write API Key
```