INSUALE RADIONICA

Vježba korištenja Insulae platforme – skripta v.1

Autori: Matea Grdić (matea.grdic@ericsson.com) i Damir Medved (damir.medved@ericsson.com) – pitanja i prijedlozi za nadopunama su dobrodošli!

Vježbe:

1. Upoznavanje sa platformom
2. Pregled podataka na pojedinim senzorima – uređivanje površine
3. Dodavanje senzora na radnu površinu
4. Ubacivanje karte i razmještanje senzora
5. Kreiranje pravila za automatizaciju
6. Pregledavanje analitičkih podataka
7. Napredno: Dubinska analiza podataka – korelacije
8. Napredno: Dodavanje ESP32 tipa senzora na platformu (samogradnje)

Vježba 1 – Upoznavanje sa platformom

Koristeći upute (https://insulae.wp.fsb.hr/upute/) – strana 5, spoji se na Insulae platformu (https://www.insulae.dynu.net/auth/login) koristeći sljedeće korisničko ime i zaporku.

Ime: opatija1@insulae.hr

Zaporka: ETKopatija1

Pregledaj osnovne elemente sučelja i izbornike. Zapamti gdje se što nalazi jer će trebati za izvođenje sljedećih vježbi.

Zadatak: Gdje se nalaze korisnički podaci? Probaj zamijeniti sliku korisnika.

Vježba 2 – Pregled podataka na pojedinim senzorima putem **widgeta**

Koristeći upute (https://insulae.wp.fsb.hr/upute/) – strana 8, pregledaj koji podaci su ti dostupni na pojedinom widgetu. Obrati pažnju na podatke koji dolaze u realnom vremenu i na povijesne podatke. Što nam je sve dostupno?

Zadatak: Pronađi i izvezi u excel podatke iz nekog senzora po izboru. Pregledaj podatke, što nam je dostupno i kako bi to mogli upotrijebiti?

Vježba 3 – Dodavanje senzora na radnu površinu

Koristeći upute (https://insulae.wp.fsb.hr/upute/) – strana 10, dodaj neki od senzora na radnu površinu. Iz padajućeg izbornika možeš odabrati bilo koji senzor (odnosno njegov widget). Pogledaj koja ima svojstva i koji podaci su dostupni. U polje IO uređaja upiši Test i neki broj.

Zadatak: Obrisati senzor koji smo dodali.

Vježba 4 – Ubacivanje karte i razmještanje senzora (samo na testnom okruženju)

Koristeći upute (https://insulae.wp.fsb.hr/upute/) – strana 10, pronađi kartu sa rasporedom senzora. Koje opcije su dostupne i u kakvom formatu treba biti tlocrt?

Zadatak: Ubaci novi tlocrt i razmjesti senzore (samo na testnom okruženju)

Vježba 5 – Kreiranje pravila za automatizaciju

Koristeći upute (https://insulae.wp.fsb.hr/upute/) – strana 11 do 12, pregledaj koja pravila su postavljena na platformi. Da li ti je jasna logika rada. Koja su ograničenja?

Zadatak: Osmisli i kreiraj novo pravilo i provjeri da li funkcionira kako je zamišljeno.

Vježba 6 – pregledavanje analitičkih podataka

Koristeći upute (https://insulae.wp.fsb.hr/upute/) – strana 18 do 21, pregledaj koje funkcionalnosti su dostupne. Zašto je važno profiliranje kućanskih uređaja i navika korisnika?

Zadatak: Utvrdi neke uzorke ponašanja korisnika na osnovi podataka koji su dostupni na grafovima. Što možemo zaključiti?

Vježba 7 – Dubinska analiza podataka - korelacije

OVO SU NAPREDNE VJEŽBE – OPCIONALNO!

Na kontrolnoj ploči mogu se pratiti senzori, klikom na određeni senzor, u desnom kutu prikazuju se podaci za određeno svojstvo koji je zabilježio senzor, podaci se mogu preuzeti klikom na tri točkice u Excel formatu koje se kasnije može koristiti za različite interpretacije i usporedbe nad podacima.

Slika 1. Kontrolna ploča za praćenje senzora na platformi Insuale

Prvi zadatak je preuzeti podatke o kvaliteti zraka u informatičkoj učionici i stanju prozora u istoj učionici. S ciljem dokazivanja hipoteze da će otvoreni prozori pridonijeti kvaliteti zraka.

timestamp	Koncentracija [ppb]
10. 05. 2022. 18:43:26	450
10. 05. 2022. 18:42:26	438
10. 05. 2022. 18:41:25	416
10. 05. 2022. 18:40:25	403
10. 05. 2022. 18:37:55	380
10. 05. 2022. 18:33:45	362
10. 05. 2022. 18:23:54	368
10. 05. 2022. 18:21:55	380
10. 05. 2022. 18:20:54	397
10. 05. 2022. 18:19:54	417
10. 05. 2022. 18:17:53	400
10. 05. 2022. 18:14:23	386

timestamp	Stanje
10. 05. 2022. 09:44:19	TRUE
04. 05. 2022. 09:40:33	FALSE
04. 05. 2022. 08:00:09	TRUE
03. 05. 2022. 16:38:02	FALSE
03. 05. 2022. 13:21:50	TRUE
29. 04. 2022. 16:34:50	FALSE
29. 04. 2022. 09:44:31	TRUE
29. 04. 2022. 09:44:31	FALSE
29. 04. 2022. 08:50:05	TRUE

Slika 2. Podaci o koncentraciji ppb-a (Tablica 1), podaci o stanju prozora (Tablica 2)

Podaci o kvaliteti zraka prikazani na Slici 2. (Tablica 1)prate se pomoću koncentracije ppb-a. Ppb se koristi za opisivanje koncentracija onečišćenja u zraku (kao volumni udio). U drugoj tablici se nalaze podaci o stanju prozora (TRUE označava da je prozor otvoren u određenom razdoblju, FALSE da je prozor zatvoren).

Dvije tablice se spoje u jednu tablicu. U stupcu 'Prozor' jadnicom je označeno kada je prozor otvoren, nulom kada je prozor zatvoren (Slika 3.). Vrijednosti se poklapaju po vremenskom razdoblju tj. Prati se da li je u trenutku kada je zabilježena koncentracija prozor bio otvoren ili nije.

Koncentracija	Prozor
771	1
753	1
676	1
752	1
722	1
772	1
787	1
774	1
760	1
798	1
744	1
723	1
764	1

Slika 3. Podaci o koncentraciji ppb-a i stanju prozora u jednoj tablici

Kako bi se napravila usporedba dvije varijable koristiti će se statistički alat RStudio.

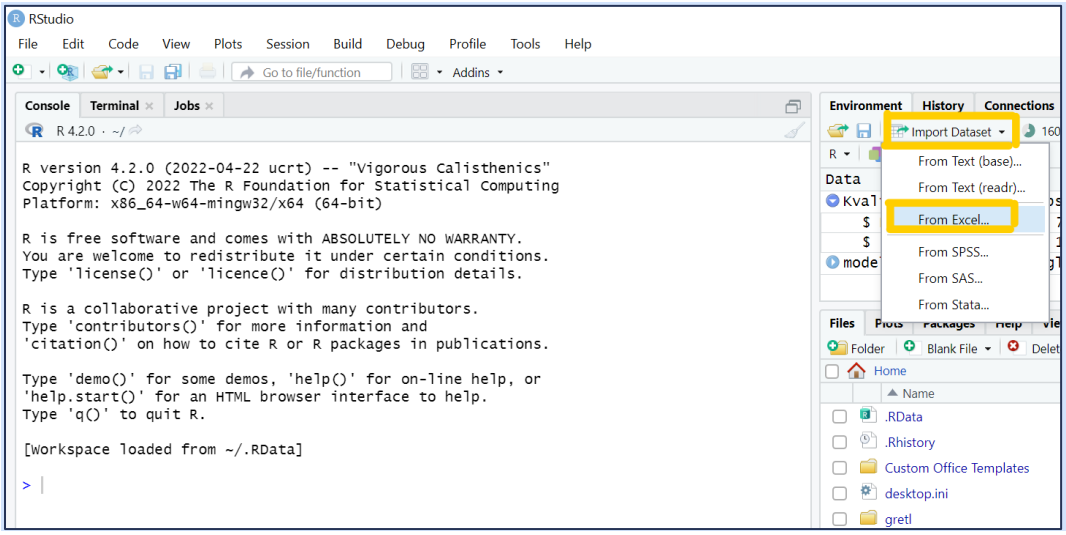
U sljedećim koracima opisan je postupak od instalacije do analize rezultata.

1. ***Instalacija R i RStudio***

R: Download R-4.2.0 for Windows. The R-project for statistical computing.

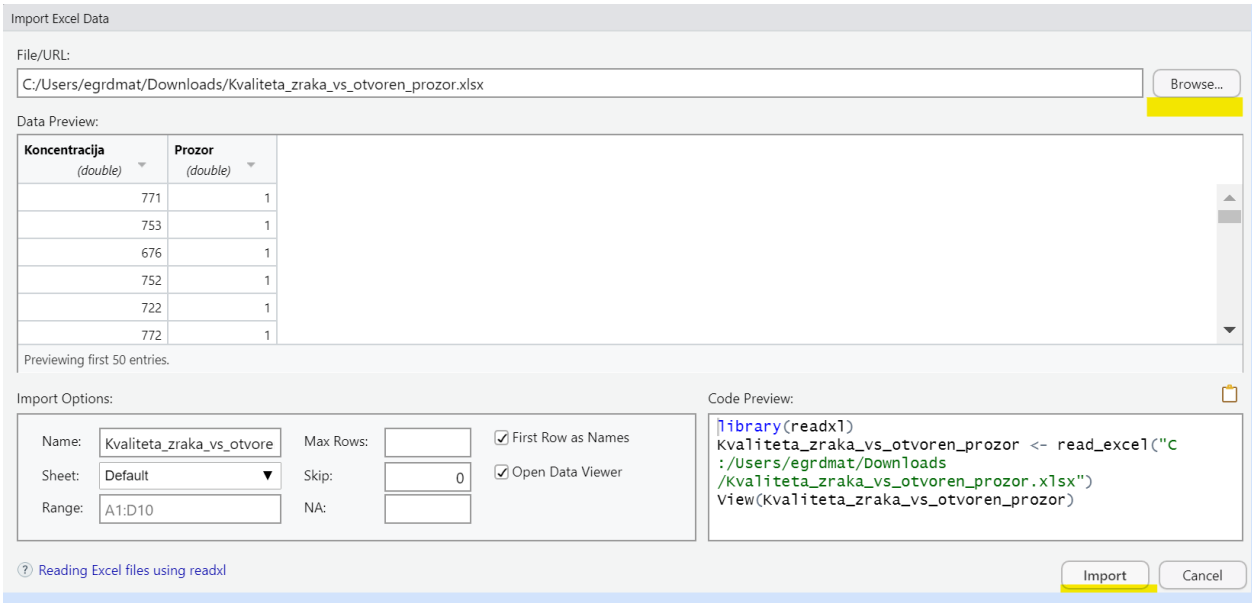
RStudio: https://www.rstudio.com/products/rstudio/download/#download

2. ***Dodavanje podataka u RStudio prikazano je na Slici 4. Potrebno je kliknuti na "Import Dataset" ->"From Excel"***



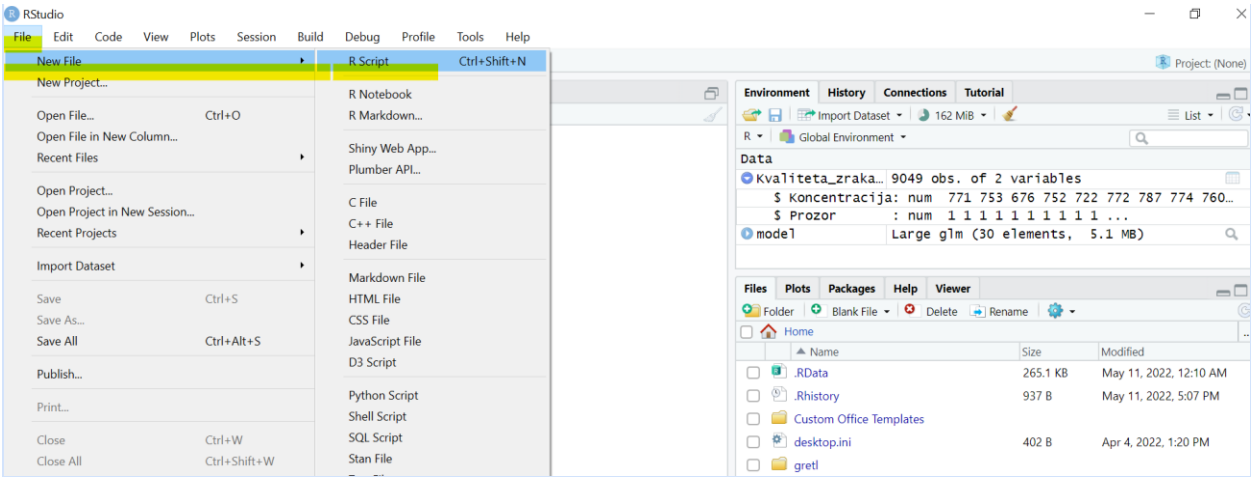
Slika 4. Dodavanje podataka u RStudio

Nakon toga podaci se učitaju s računala kao što je prikazano na Slici 5.



Slika 5. Odabir datoteke s računala

Nakon učitanih podataka potrebno je izraditi novu skriptu kao na Slici 6.



Slika 6. Kreiranje nove skripte u RStudioy

Kako bi se napravila usporedba dvije varijable: izrađen je t-test. T-test je statistički test koji se koristi za usporedbu srednjih vrijednosti dviju skupina. Često se koristi u testiranju hipoteza kako bi se utvrdilo ima li proces ili tretman zapravo učinak na populaciju od interesa ili se dvije skupine razlikuju jedna od druge.

Sljedećom formulom prikazano je kreiranje t-testa. Za usporedbu koncentracije na stanje prozora (otvoren/ zatvoren). Rezultati su prikazani na Slici 7.

t.test (Kakvoca ~ Prozor , var.equal=TRUE, data= Prozor_vs_kakvoca_inf)

t.test funkcija za t test

var.equal TRUE, predstavlja da se koristi skupna procjena varijance prema zadanim postavkama

Koncentracija ~ Prozor , uspoređuju se dvije varijable koncentracija i prozor (nazivi stupca)

data= Kvaliteta_zraka_vs_otvoren_prozor – data='ime tablice'

```
        Two Sample t-test

data:  Kakvoca by Prozor
t = 64.767, df = 3849, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 540.2855 574.0168
sample estimates:
mean in group 0 mean in group 1
      1336.1794       779.0283
```

Slika 7. Rezultati t-testa za podatke koncentracija ppb-a i stanje prozora u informatičkoj učionici

Postoji statistički značajna razlika u koncentraciji zraka obzirom na stanje prozora (Zatvoren/ Otvoren), t(3849)=27, p<0,05, u informatičkoj učionici.

Kada je prozor zatvoren koncentracija je značajno viša (aritmetička sredina =1336.18.01) u odnosu na koncentraciju zraka kada je prozor bio otvoren (aritmetička sredina=779.03) u informatičkoj učionici.

Postoji statistički značajna razlika u koncentraciji zraka obzirom na stanje prozora (Zatvoren/ Otvoren), t(8782)=34, p<0,05, u tehničkoj učionici.

```
        Two Sample t-test

data:  Koncentracija by Prozor
t = 34.005, df = 8782, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 649.6189 729.0965
sample estimates:
mean in group 0 mean in group 1
      1134.4585       445.1008
```

Slika 8. Rezultati t-testa za podatke koncentracija ppb-a i stanje prozora u tehničkoj učionici

Kada je prozor zatvoren koncentracija je značajno viša (aritmetička sredina =1134.45) u odnosu na koncentraciju zraka kada je prozor bio otvoren (aritmetička sredina=445.1) u tehničkoj učionici.

U sljedećem primjeru promatrati će se dali pokret u učionici utječe na koncentraciju ppb-a u učionici. Također, kao i u prethodno zadatku izrađen je t-test. Rezultati su prikazani na Slici 9.

t.test (Koncentracija ~ Pokret , var.equal=TRUE, data= Pokret_vs_koncentracija)

```
        Two Sample t-test

data:  Koncentracija by Pokret
t = 0.21759, df = 1088, p-value = 0.8278
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
95 percent confidence interval:
 -149.4402  186.7188
sample estimates:
mean in group 0 mean in group 1
       1292.113        1273.474
```

Slika 9. Rezultati t-testa za podatke koncentracija ppb-a i pokret u informatičkoj učionici

p- vrijednost je značajno veća od 0.05 što ukazuje da značaj pokreta u učionici ne igra ulogu u kvaliteti zraka.

Dodatna pitanja i zadaci za naprednije učenike: Analiza i vizualizacija podataka u programu RStudio

1. Koliko ima opažanja, a koliko varijabli u tablici 'Kvaliteta_zraka_vs_otvoren_prozor'?

nrow(Kvaliteta_zraka_vs_otvoren_prozor)

[1] 9049 opažanja

ncol(Kvaliteta_zraka_vs_otvoren_prozor)

[1] 2 varijable

2. Koja je aritmetička sredina varijable 'Koncentracija' u tablici 'Kvaliteta_zraka_vs_otvoren_prozor'?

mean(Kvaliteta_zraka_vs_otvoren_prozor$Koncentracija)

[1] aritmetička sredina iznosi 1293.219

3. Koja je najviša vrijednost koncentracije ppb-a u zadanoj tablici?

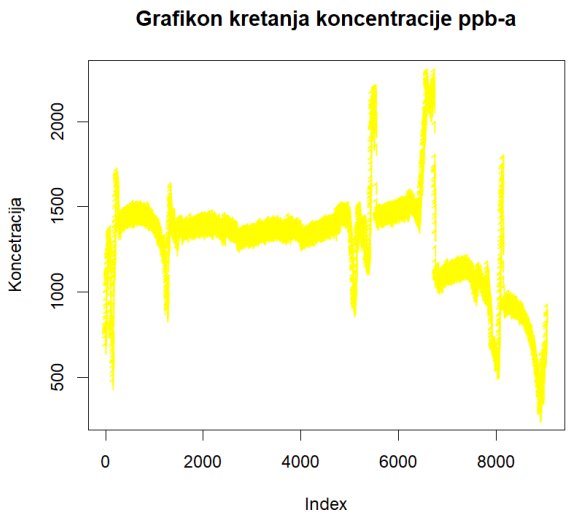
najvisavrijednost_koncentracije<-max(Kvaliteta_zraka_vs_otvoren_prozor$Koncentracija)

najvisavrijednost_koncentracije

[1] najviša vrijednost iznosi 2279

4. Izradite grafikon koji objašnjava kretanje varijable koncentracije ppb-a, neka se grafikon zove "Grafikon kretanja koncentracije ppb-a" i neka bude žute boje, neka se y-koordinata zove „Koncentracija“

```
plot(Kvaliteta_zraka_vs_otvoren_prozor$Koncentracija, main="Grafikon kretanja koncentracije ppb-a", col="yellow", pch="15", ylab="Koncetracija")
```



5. Izradite novu tablicu u kojoj će se izdvojiti podaci samo kada je prozor otvoren. Koja je prosječna razina koncentraciji ppb-a kada je prozor otvoren?

```
kvaliteta_otvoren_prozor<-
Kvaliteta_zraka_vs_otvoren_prozor[Kvaliteta_zraka_vs_otvoren_prozor$Prozor==1,]

 mean(kvaliteta_otvoren_prozor$Koncentracija)
```

[1] Prosječna razina koncentracije ppb-a kada je prozor otvoren iznosi 945.1318

6. Izradite novu tablicu u kojoj će se izdvojiti podaci samo kada je prozor zatvoren. Koja je prosječna razina koncentraciji ppb-a kada je prozor zatvoren?

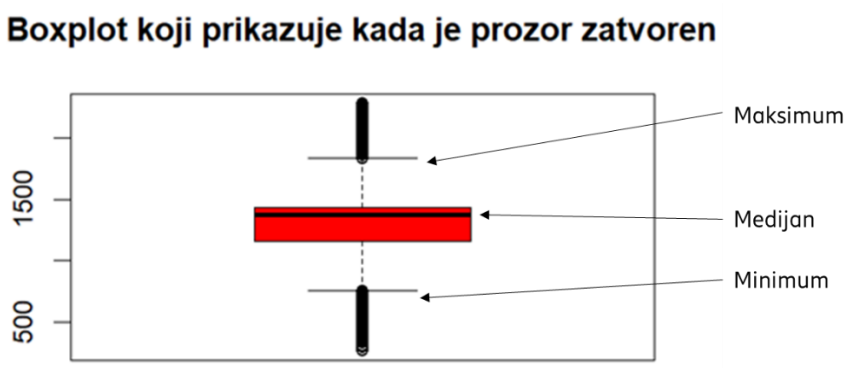
```
kvaliteta_zatvoren_prozor<-
Kvaliteta_zraka_vs_otvoren_prozor[Kvaliteta_zraka_vs_otvoren_prozor$Prozor==0,]

mean(kvaliteta_zatvoren_prozor$Koncentracija)
```

[1] Prosječna razina koncentracije ppb-a kada je prozor zatvoren iznosi 1311.01

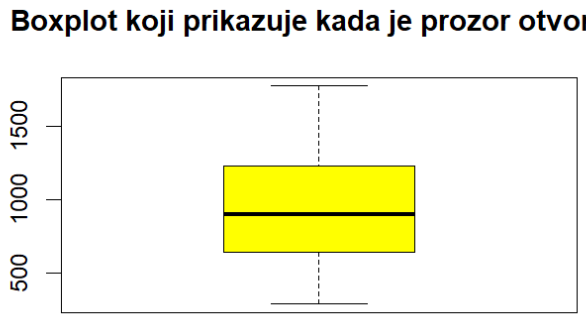
7. Izradite Kutijasti dijagram (Box plot) koji prikazuje distribuciju kada je prozor otvoren i zatvoren. Kada je prozor zatvoren neka dijagram bude crvene boje a kada je otvoren neka bude žuti.

**Kutijasti grafikoni koriste se za prikaz distribucije brojčanih vrijednosti podataka, osobito kada ih želite usporediti između više grupa. Napravljeni su tako da na prvi pogled pruže informacije visoke razine, nudeći općenite informacije o skupini podataka o simetriji, iskrivljenosti, varijanci i odstupnicima. Lako je vidjeti gdje je glavnina podataka i napraviti tu usporedbu između različitih skupina.

boxplot(kvaliteta_zatvoren_prozor$Koncentracija, main="Boxplot koji prikazuje kada je prozor zatvoren", col="red")

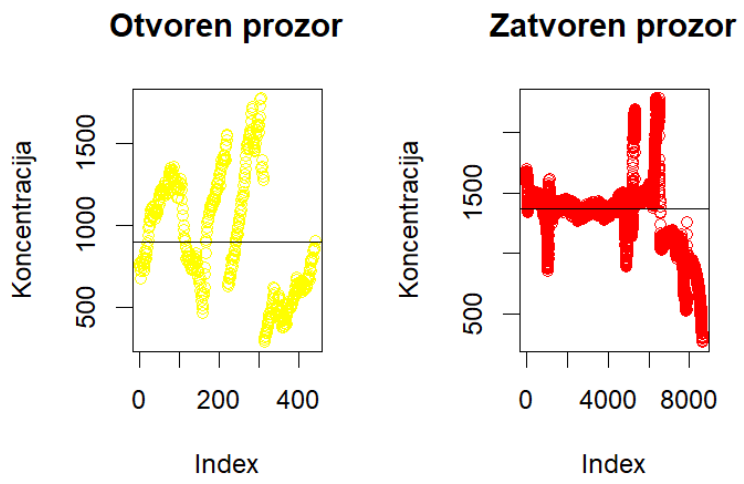


boxplot(kvaliteta_otvoren_prozor$Koncentracija, main="Boxplot koji prikazuje kada je prozor otvoren", col="yellow")



8. Prikažite na usporednom grafikonu (jedan do drugoga) odnos kada je prozor zatvoren i kada je otvoren promatrajući koncentraciju ppb-a.

```
par(mfrow=c(1,2))

plot(kvaliteta_otvoren_prozor$Koncentracija, col="yellow", main="Otvoren prozor",ylab="Koncentracija")

abline(h=median(kvaliteta_otvoren_prozor$Koncentracija))

plot(kvaliteta_zatvoren_prozor$Koncentracija, col="red", main="Zatvoren prozor",ylab="Koncentracija")

abline(h=median(kvaliteta_zatvoren_prozor$Koncentracija))
```



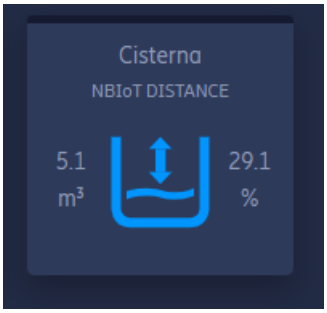
Vježba 8 – Dodavanje ESP32 tipa senzora na platformu (samogradnje)

OVO SU NAPREDNE VJEŽBE – OPCIONALNO!

Na Insulae platformu moguće je dodavati i senzore koji su namijenjeni edukaciji i samogradnji, a bazirani su na Arduino, ESP32 ili 8266 tehnologijama. Kodiranje se izvodi u Arduino IDE ili Visual Studio Code sa Platformio plugin-om (preferabilno).

U nastavku donosimo jednostavan primjer laserskog daljinomjera (primjer uređaja: https://www.sparkfun.com/products/15179) koji dostavlja podatke u Insulae platformu. Može se koristiti za mjerenje visine tekućine u rezervoaru ili cisterni. Riječ je o ToF (time-of-flight) LiDAR uređaju koji svoj rad temelji na odašiljanju impulsa infracrvenog svijetla, te mjerenju refleksije i proteklog vremena. Jednostavnije rečeno, ovaj senzor šalje svjetlosne impulse koji su nevidljivi golim okom, te mjeri koliko je vremena bilo potrebno da svijetlost dođe do objekta za koji tražimo udaljenost i vrati se natrag do senzora. Ta pojava naziva se refleksija.

Načelno se isti kod uz minimalne modifikacije može iskoristiti za prikupljanje podataka sa bilo kakvog senzora – temperature, buke, svjetlosti i sl.



U nastavku donosimo kratko objašnjenje funkcionalnosti:

1. **Dio – inicijalizacija**

U ovom koraku definiramo sve potrebne biblioteke funkcija, kao i početne postavke mreže na koju se spajamo. Biblioteke funkcija uvrštavaju sa „#include <IME BIBLIOTEKE.h>“ i te biblioteke nam služe kako ne bismo svaki korak ili funkciju morali ručno implementirati, nego možemo iskoristiti ono što je netko prije već napravio. Zatim slijedi deklariranje globalnih varijabli sa kojima naš senzor radi, a u ovom slučaju to su 3 varijable – varijabla u koju će se pohranjivati vrijednost izmjerena udaljenost objekta, varijabla u koju će se pohranjivati snaga signala i varijabla za pohranu temperature senzora. Nakon toga definiramo Wi-Fi mrežu na koju ćemo se spajati tako da upišemo ime mreže i lozinku. Ukoliko ovdje upišemo krive podatke, uređaj se neće spojiti na mrežu, te se program neće niti pokrenuti – to je sigurnosni mehanizam da nas uređaj ne bi zavarao i počeo raditi mjerenja, a da pritom ne isporučuje podatke na odredišni server.

```
#include <PubSubClient.h>
#include <WiFiClientSecure.h>

#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <TFMPlus.h>  // Include TFMini Plus Library v1.4.1
TFMPlus tfmP;         // Create a TFMini Plus object
HardwareSerial mySerial(2);     //ESP32 pins GPIO16/GPIO17

int16_t tfDist = 0;    // Distance to object in centimeters
int16_t tfFlux = 0;    // Strength or quality of return signal
int16_t tfTemp = 0;    // Internal temperature of Lidar sensor chip

/* change it with your ssid-password */
const char* ssid = "ime mreze";
const char* password = "lozinka";

#if defined(ESP32)
#include <WiFi.h>z

char* serverHostname = "111.111.111.111";
```

2. **Digitalni certifikat – omogućava pristup na Insulae platformu (treba zatražiti od Ericssona – ovo je samo primjer!)**

Digitalni certifikat je jedan od slojeva sigurnosti. Naime, želimo da naše podatke može pročitati samo naš server, i nitko drugi. Budući da se podaci šalju prvo na lokalnu mrežu, pa zatim putem interneta do odredišnog servera, te podatke može presresti netko tko prati promet na našoj mreži, pročitati ih pa čak i modificirati, čime bismo na drugoj strani dobivali netočne podatke. Kako bismo to spriječili, dodan je sigurnosni certifikat koji šifrira (kriptira) poruku prije nego ona napusti naš Arduino ili ESP uređaj. Napadač koji prisluškuje promet sada ne zna što piše u poruci, te ju ne može niti izmijeniti.

Odredišni server zna kako dekodirati poruku, tj. ima drugi dio ključa (asimetrična kriptografija) te uz pomoć njega može dekodirati poruku i ispravan podatak zapisati u bazu podataka.

Važno je znati da ovaj certifikat može biti javno vidljiv zato što je riječ o tzv. javnom ključu. Svi koji žele slati podatke na naš server moraju ih kriptirati koristeći taj ključ. Kako bi se podaci ponovno mogli pročitati potreban je tzv. privatni ključ, koji mora ostati tajan, odnosno ne smije ga nikada saznati nitko tko nema prava pristupa našem serveru. Na ovaj način postigli smo visoku razinu sigurnosti, te smo omogućili apsolutno svima da na siguran način razmjenjuju poruke sa našim serverom bez potrebe da se upisuju bilo kakve lozinke.

```
// Root CA
const static char* ca_cert PROGMEM = \
    "-----BEGIN CERTIFICATE-----\n" \
    "MIIFdzCCBF+gAwIBAgIQE+oocFv07O0MNmMJgGFDNjANBgkqhkiG9w0BAQwFADBv\n" \
    "MQswCQYDVQQGEwJTRTEUMBIGA1UEChMLQWRkVHJ1c3QgQUIxJjAkBgNVBAsTHUFk\n" \
    "ZFRydXN0IEV4dGVybmFsIFRUUCBOZXR3b3JrMSIwIAYDVQQDExlBZGRUcnVzdCBF\n" \
    "eHRlcm5hbCBDQSBSb290MB4XDTAwMDUzMDEwNDgzOFoXDTIwMDUzMDEwNDgzOFow\n" \
    "gYgxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpOZXcgSmVyc2V5MRQwEgYDVQQHEwtK\n" \
    "ZXJzZXkgQ2l0eTEeMBwGA1UEChMVVGhlIFVTRVJUUlVTVCBOZXR3b3JrMS4wLAYD\n" \
    "VQQDEyVVU0VSVHJ1c3QgUlNBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIICIjAN\n" \
    "BgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAgBJlFzYOw9sIs9CsVw127c0n00yt\n" \
    "UINh4qogTQktZAnczomfzD2p7PbPwdzx07HWezcoEStH2jnGvDoZtF+mvX2do2NC\n" \
    "tnbyqTsrkfjib9DsFiCQCT7i6HTJGLSR1GJk23+jBvGIGGqQIjy8/hPwhxR79uQf\n" \
    "jtTkUcYRZ0YIUcuGFFQ/vDP+fmyc/xadGL1RjjWmp2bIcmfbIWax1Jt4A8BQOujM\n" \
    "8Ny8nkz+rwWWNR9XWrf/zvk9tyy29lTdyOcSOk2uTIq3XJq0tyA9yn8iNK5+O2hm\n" \
    "AUTnAU5GU5szYPeUvlM3kHND8zLDU+/bqv50TmnHa4xgk97Exwzf4TKuzJM7UXiV\n" \
    "Z4vuPVb+DNBpDxsP8yUmazNt925H+nND5X4OpWaxKXwyhGNVicQNwZNUMBkTrNN9\n" \
    "N6frXTpsNVzbQdcS2qlJC9/YgIoJk2KOtwbPJYjNhLixP6Q5D9kCnusSTJV882sF\n" \
    "qV4Wg8y4Z+LoE53MW4LTTLPtW//e5XOsIzstAL81VXQJSdhJWBp/kjbmUZIO8yZ9\n" \
    "HE0XvMnsQybQv0FfQKlERPSZ51eHnlAfV1SoPv10Yy+xUGUJ5lhCLkMaTLTwJUdZ\n" \
    "+gQek9QmRkpQgbLevni3/GcV4clXhB4PY9bpYrrWX1Uu6lzGKAgEJTm4Diup8kyX\n" \
    "HAc/DVL17e8vgg8CAwEAAaOB9DCB8TAfBgNVHSMEGDAWgBStvZh6NLQm9/rEJlTv\n" \
    "A73gJMtUGjAdBgNVHQ4EFgQUU3m/WqorSs9UgOHYm8Cd8rIDZsswDgYDVR0PAQH/\n" \
    "BAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wEQYDVR0gBAowCDAGBgRVHSAAMEQGA1Ud\n" \
    "HwQ9MDswOaA3oDWGM2h0dHA6Ly9jcmwudXNlcnRydXN0LmNvbS9BZGRUcnVzdEV4\n" \
    "dGVybmFsQ0FSb290LmNybDA1BggrBgEFBQcBAQQpMCcwJQYIKwYBBQUHMAGGGWh0\n" \
    "dHA6Ly9vY3NwLnVzZXJ0cnVzdC5jb20wDQYJKoZIhvcNAQEMBQADggEBAJNl9jeD\n" \
    "lQ9ew4IcH9Z35zyKwKoJ8OkLJvHgwmp1ocd5yblSYMgpEg7wrQPWCcR23+WmgZWn\n" \
    "RtqCV6mVksW2jwMibDN3wXsyF24HzloUQToFJBv2FAY7qCU2DrvMKnXduXBBP3zQ\n" \
    "YzYhBx9G/2CkkeFnvN4ffhkUyWNnkepnB2u0j4vAbkN9w6GAbLIevFOFfdyQoaS8\n" \
    "Le9Gclc1Bb+7RrtubTeZtv8jkpHGbkD4jylW6l/VXxRTrPBPYer3IsynVgviuDQf\n" \
    "Jtl7GQVoP7o81DgGotPmjw7jtHFtQELFhLRAlSv0ZaBIefYdgWOWnU914Ph85I6p\n" \
    "0fKtirOMxyHNwu8=\n" \
    "-----END CERTIFICATE-----\n";
```

Slika 10: Primjer digitalnog certifikata

3. Inicijalizacija komunikacija (treba zatražiti od Ericssona – ovo je samo primjer!)

Za komunikaciju sa serverom koristimo MQTT tehnologiju. Da bi sve funkcioniralo, za MQTT je potrebno definirati tzv. topic na koji se šalje. Nama su potrebna dva, jedan za komunikaciju prema serveru, i jedan za komunikaciju od servera prema našem uređaju. Ovaj topic mora biti definiran sukladno nekim pravilima na odredišnom serveru. Osim toga potrebno je podesiti i **jedinstveno ime senzora** kako bi naš sustav mogao razlikovati tko mu šalje podatke.

```
/* create an instance of WiFiClientSecure */
WiFiClientSecure espClient;
PubSubClient client(espClient);

/* topics */
char* downlink = "lidar/eb-b827eb2668b80022/downlink";
char* uplink = "json/nb-esp32-lidar-test/etk-operator";
String sensorId = "esp32-lidar-test"; // SENSOR_ID for registered gateway (GATEWAY_ID)
String clientId = "ESP32Lidar-"; /* MQTT client ID (will add random hex suffix during setup) */
```

4. Funkcija za prikupljanje podataka sa senzora – primjer daljinomjera

U ovoj funkciji odvija se mjerenje udaljenosti do objekta. Funkcija je napravljena tako da poziva mjerenje 20 puta u jednoj sekundi (svakih 50 milisekundi senzor odašilje svjetlosni impuls). Budući da je ovaj kod pisan u **Visual Studio Code** programu, on automatski boja u zeleno sve linije koje nisu aktivne prilikom izvršavanja koda, odnosno linije koje su zakomentirane. U tom komentaru mogu stajati neke upute ili pojašnjenje same linije, kao što je primjer na slici iznad, a komentiranje može biti i način isključivanja neke linije koda iz izvršavanja. Tako možemo vidjeti da smo mi zakomentirali dio gdje senzor mjeri kvalitetu signala i temperaturu, odnosno zanima nas samo udaljenost do objekta. Premda smo gore definirali varijable za sve parametre, mi ćemo iskoristiti samo varijablu za pohranjivanje izmjerene udaljenosti, dok će sve ostale trajno imati vrijednost 0. Ako nam nekada zatrebaju i drugi parametri, potrebno je samo maknuti oznaku komentara ispred odgovarajućeg programskog koda, i senzor će početi mjeriti i druge podatke. Nakon što su podaci izmjereni, formatirano ih u JSON poruku (po pravilima koje imamo na našem serveru).

```
void sensorAction() {

    delay(50);   // Loop delay to match the 20Hz data frame rate
    if( tfmP.getData( tfDist, tfFlux, tfTemp)) // Get data from the device.
    {
      printf( "Dist:%04icm ", tfDist);   // display distance,
      //printf( "Flux:%05i ",   tfFlux);   // display signal strength/quality,
      //printf( "Temp:%2i C",  tfTemp);   // display temperature,
      printf( "\r\n");                   // end-of-line.
    }
    else                  // If the command fails...
    {
      tfmP.printFrame();  // display the error and HEX dataa
    }

    StaticJsonDocument<256> data;
    data.createNestedObject("header");
    JsonObject body = data.createNestedObject("body");
    JsonObject sensor = body.createNestedObject(sensorId);
    //JsonObject resources = sensor.createNestedObject("payload_value");
    sensor["measuredValue"] = tfDist;
    sensor["txPower"] = tfFlux;
    sensor["temperature"] = tfTemp;
    String json_string;
    serializeJson(data, json_string);
    client.publish(uplink, json_string.c_str());
    delay(10000);
}
```

5. Funkcije za komunikaciju – uspostava i održavanje veze sa platformom

Ovdje možemo vidjeti da je jako puno programskog koda zakomentirano, odnosno da se ne izvršava, Riječ je o programskom kodu koji nam je olakšavao pisanje i testiranje, ali jednom kad znamo da program radi više nam nije potreban. Uređaj provjerava da li ima stabilnu konekciju sa našim serverom, te ukoliko dođe do pucanja veze, ona će se opet pokušati uspostaviti za 5 sekundi, i tako sve dok veza ne bude uspostavljena.

```
void receivedCallback(char* topic, byte* payload, unsigned int length) {
  //Serial.print("Message received: ");
  //Serial.println(topic);

  //Serial.print("payload: ");
  for (int i = 0; i < length; i++) {
    //Serial.print((char)payload[i]);
  }
  Serial.println();
  if ((char)payload[0] == '0') {
    //sensorAction();
  //  digitalWrite(RELAY, HIGH);
  //  delay (2000);
  //  digitalWrite(RELAY, LOW);
  //  sensorAction();
  }
}

void mqttconnect() {
  while (!client.connected()) {
    Serial.print("MQTT connecting ...");
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      //client.subscribe(downlink);
    } else {
      Serial.print("failed, status code =");
      Serial.print(client.state());
      Serial.println("try again in 5 seconds");
      delay(5000);
    }
  }
}
```

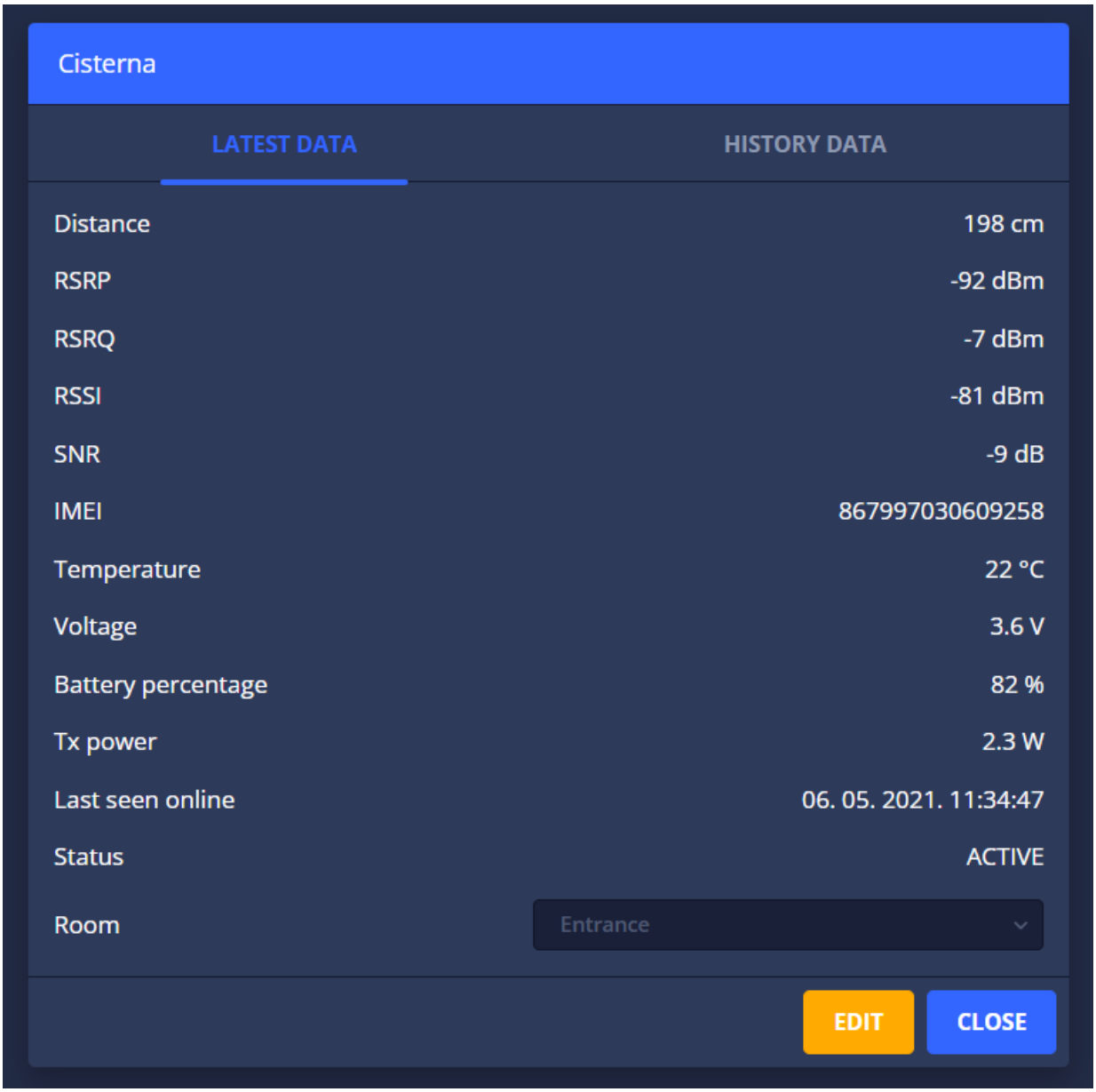
6. Glavna programska petlja I. dio

Do sada smo samo definirali potrebno radnje i parametre, a sada ih pozivamo i pokrećemo funkcije. Prvo kažemo Arduinu da se spoji na Wi-Fi sa parametrima koje smo mu dali na prvoj slici, zatim provjeravamo da li je konekcija bila uspješna. Ukoliko jest, ispisujemo lokalnu IP adresu našeg Arduina, te mu kažemo da sada kada ima Internet pokuša uspostaviti vezu sa našim serverom za prikupljanje podataka. I prije bilo kakve druge akcije, neka Arduino resetira senzor za mjerenje udaljenosti, kako bismo bili sigurni da senzor nema u sebi pohranjenu neku vrijednost koju je ranije izmjerio.

```
void setup() {
  Serial.begin(9600);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

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

  espClient.setCACert(ca_cert);
  client.setServer(serverHostname, 8883);
  client.setCallback(receivedCallback);

  mySerial.begin(115200);  // Initialize TFMPLus device serial port.
  delay(20);               // Give port time to initalize
  tfmP.begin( &mySerial);   // Initialize device library object and...
                            // pass device serial port to the object.

  // - - Perform a system reset - - - - - - - - - - -
  printf( "System reset: ");
  if( tfmP.sendCommand( SYSTEM_RESET, 0))
  {
    printf( "passed.\r\n");
  }
  else tfmP.printReply();
  delay(500);  // added to allow the System Rest enough time to complete
```

7. Glavna programska petlja II. dio

Ovdje vidimo određene sistemske stvari koje su pomalo specifične za ovaj tip senzora. Kod programiranja za Arduino, sve ono što želimo da se stalno izvršava stavljamo unutar funkcije *loop()*. Tako smo unutar naše funkcije loop() stavili da se uvijek poziva funkcija koja provjerava stabilnost veze i funkcija koja mjeri udaljenost do objekta.

```
  // - - Perform a system reset - - - - - - - - - - -
  printf( "System reset: ");
  if( tfmP.sendCommand( SYSTEM_RESET, 0))
  {
      printf( "passed.\r\n");
  }
  else tfmP.printReply();
  delay(500);  // added to allow the System Rest enough time to complete

// - - Display the firmware version - - - - - - - - -
  printf( "Firmware version: ");
  if( tfmP.sendCommand( OBTAIN_FIRMWARE_VERSION, 0))
  {
      printf( "%1u.", tfmP.version[ 0]); // print three single numbers
      printf( "%1u.", tfmP.version[ 1]); // each separated by a dot
      printf( "%1u\r\n", tfmP.version[ 2]);
  }
  else tfmP.printReply();
  // - - Set the data frame-rate to 20Hz - - - - - - - -
  printf( "Data-Frame rate: ");
  if( tfmP.sendCommand( SET_FRAME_RATE, FRAME_20))
  {
      printf( "%2uHz.\r\n", FRAME_20);
  }
  else tfmP.printReply();
  delay(500);            // And wait for half a second.
}

void loop() {
  if (!client.connected()) {
    mqttconnect();
  }
    client.loop();

//add custom functions bellow this line
    sensorAction();
}
```

Ukoliko je sve dobro napravljeno, senzor bi trebao mjeriti podatke, formatirati ih po ranije postavljenim i opisanim pravilima za JSON format, te ih uspješno poslati na server koji sluša na drugoj strani. S druge strane, ukoliko je sve bilo dobro, server bi trebao uspješno pročitati podatke koje mu je senzor poslao te ih u gotovo realnom vremenu prikazati kao rezultat na korisničkom sučelju.