



# MQTT protokol

# MQTT protokol

U današnjem međusobno povezanom svijetu, gdje uređaji neprimjetno komuniciraju kako bi olakšali automatizaciju i razmenu podataka, razumijevanje MQTT protokola postaje sve vrednije.

Bez obzira da li ste programer koji se bavi IoT projektima ili ste jednostavno radoznali kako uređaji razgovaraju jedni sa drugima preko mreža, ovaj vodič će vas provesti kroz osnove MQTT-a, njegove ključne koncepte i njegove praktične primjene.



# MQTT protokol

- ▶ Što je MQTT?
- ▶ Zašto je MQTT veoma popularan za IoT?
- ▶ Kako MQTT radi?
- ▶ MQTT radni tok.
- ▶ Početak rada sa MQTT: Brzi vodič.

# Što je MQTT?

MQTT (Message Queuing Telemetry Transport) je lagan protokol za razmjenu poruka zasnovan na objavljivanju i preplati.

Dizajniran je za uređaje sa ograničenim resursima i niske propusnosti, velikih kašnjenja ili nepouzdane mreže.

Široko se koristi u aplikacijama Interneta stvari (IoT), obezbjeđujući efikasnu komunikaciju između senzora, aktuatora i drugih uređaja.

# Zašto je MQTT veoma popularan za IoT?

MQTT važi za jedan od najboljih IoT protokola zbog svojih jedinstvenih osobina i mogućnosti, prilagođenih specifičnim potrebama IoT sistema.

Neki od ključnih razloga uključuju:

- ▶ **Lagan:** IoT uređaji često su ograničeni u smislu procesorske snage, memorije i potrošnje energije. MQTT-ovo minimalno opterećenje i mala veličina paketa čine ga idealnim za ove uređaje, budući da troši manje resursa, omogućujući učinkovitu komunikaciju čak i sa/između uređaja s ograničenim mogućnostima.
- ▶ **Pouzdan:** IoT mreže mogu imati visoke latencije ili nestabilne veze. MQTT osigurava pouzdanu isporuku poruka čak i u izazovnim uslovima, što ga čini prikladnim za IoT aplikacije.
- ▶ **Sigurnost komunikacije:**
  - ▶ **Sigurnost:** Sigurnost je ključna u IoT mrežama, jer često prenose osjetljive podatke. MQTT podržava Transport Layer Security (TLS) i Secure Sockets Layer (SSL) enkripciju, osiguravajući povjerljivost podataka tokom prenosa. Dodatno, pruža mehanizme provjere autentičnosti i autorizacije putem akreditiva korisničkog imena/lozinke ili certifikata klijenta, štiteći pristup mreži i njezinim resursima.
  - ▶ **Dvosmjernost:** MQTT-ov model objavljivanja i pretplate omogućuje besprekornu dvosmjernu komunikaciju između uređaja. Klijenti mogu objavljivati poruke na teme i pretplatiti se na primanje poruka o određenim temama, omogućujući učinkovitu razmjenu podataka u različitim IoT sistemima bez direktnog povezivanja između uređaja. Ovaj model takođe pojednostavljuje integraciju novih uređaja, osiguravajući jednostavnu skalabilnost.

# Zašto je MQTT veoma popularan za IoT?

- ▶ **Sigurnost komunikacije:**
  - ▶ **Kontinualne sesije s praćenjem stanja:** MQTT omogućuje klijentima održavanje sesija s praćenjem stanja s brokerom, omogućujući sistemu da pamti preplate i neisporučene poruke čak i nakon prekida veze. Klijenti također mogu, tokom povezivanja, odrediti interval održavanja, što od brokera traži da povremeno provjerava status veze. Ako se veza izgubi, broker pohranjuje neisporučene poruke (ovisno o QoS nivou) i pokušava ih isporučiti kada se klijent ponovno poveže. Ova osobina osigurava pouzdanu komunikaciju i smanjuje rizik od gubitka podataka.
  - ▶ **Podrška za IoT s velikim brojem uređaja:** IoT sistemi često uključuju veliki broj uređaja, zahtijevajući protokol koji može podnijeti implementacije velikih razmjera. MQTT-ova laganost, niska potrošnja propusnog opsega i efikasno korištenje resursa, čine ga prikladnim za velike IoT aplikacije. Obrazac objavljivanje-preplata omogućuje efikasno skaliranje MQTT-a, jer odvaja pošiljaoca i primaoca, smanjujući mrežni saobraćaj i korištenje resursa.
  - ▶ **Jezična podrška:** IoT sistemi često uključuju uređaje i aplikacije razvijene pomoću različitih programskih jezika. MQTT-ova široka jezična podrška omogućuje jednostavnu integraciju s više platformi i tehnologija, podsstičući besprekornu komunikaciju i interoperabilnost u različitim IoT sistemima. MQTT se može koristiti u C++-u, PHP-u, Node.js, Python, Golang i drugim programskim jezicima.

# Kako MQTT radi?

Da biste razumjeli kako MQTT radi, prvo se morate upoznati s konceptima:

- ▶ MQTT klijent,
- ▶ MQTT broker,
- ▶ metoda objave-pretpiske,
- ▶ teme i
- ▶ QoS.

# Kako MQTT radi?

## MQTT klijent:

Svaka aplikacija ili uređaj koji pokreće biblioteku MQTT klijenta je MQTT klijent.

Na primjer:

- ▶ aplikacija za razmjenu trenutnih poruka, koja koristi MQTT, je klijent,
- ▶ različiti senzori koji koriste MQTT za dojavu podataka su klijent,
- ▶ razni alati za testiranje MQTT također su klijent.

# Kako MQTT radi?

## MQTT broker:

MQTT Broker upravlja klijentskim vezama, prekidom, zahtevima za pretplatu i otkazivanje pretplate i porukama za rutiranje.

Moćan MQTT broker može da podrži brojne veze i protok poruka na milionskom nivou, pomažući provajderima IoT usluga da se fokusiraju na posao i brzo kreiraju pouzdanu MQTT aplikaciju.

[10 Free Public MQTT Brokers\(Private & Public\) - Mntolia.com](https://www.mntolia.com/10-free-public-mqtt-brokers-private-public)

# Kako MQTT radi?

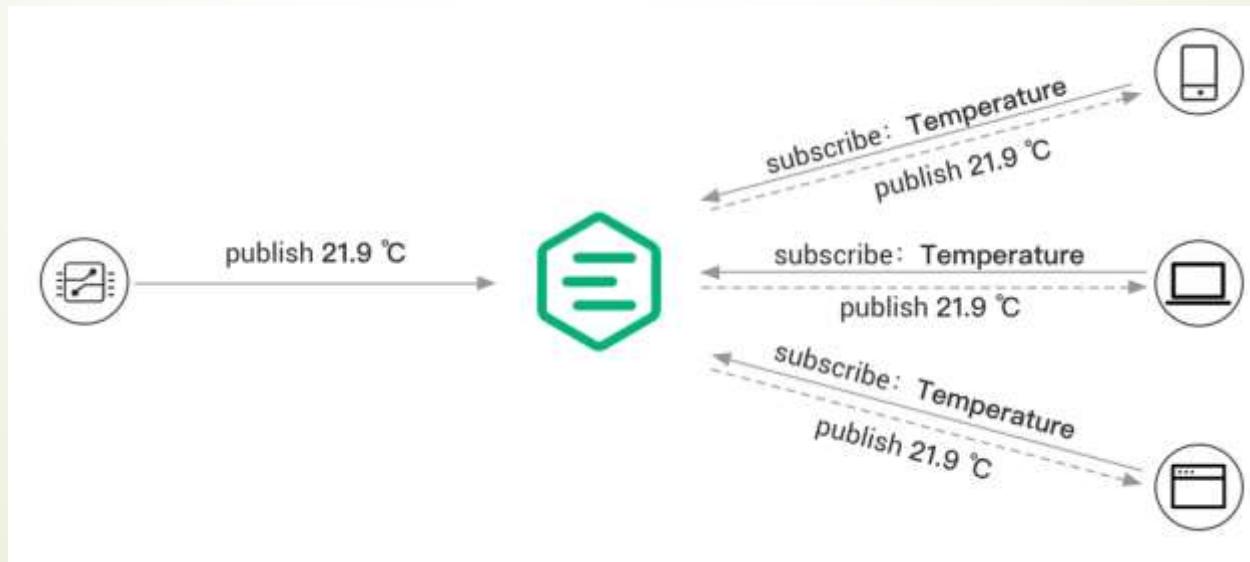
## Obrazac objava-preplata:

Obrazac objava-preplata se razlikuje od obrasca klijent-server po tome što odvaja klijenta koji šalje poruke (izdavača) od klijenta koji prima poruke (pretplatnika). Izdavači i pretplatnici ne moraju da uspostavljaju direktnu vezu, a MQTT Broker je odgovoran za rutiranje i distribuciju svih poruka.

# Kako MQTT radi?

## Obrazac objava-preplata:

Dijagram prikazuje MQTT proces objavljivanja/preplate. Senzor temperature se povezuje sa MQTT serverom kao klijent i objavljuje podatke o temperaturi u temi (npr. Temperatura), a server prima poruku i prosleđuje je klijentu koji je preplaćen na temu Temperatura.



# Kako MQTT radi?

## Teme:

MQTT protokol usmjerava poruke na osnovu teme. Tema razlikuje hijerarhiju kosom crtom /, što je slično URL putanjama, na primjer:

razgovor/soba/1

senzor/10/temperatura

senzor/+/temperatura

MKTT tema podržava sljedeće džoker znakove: + i #.

+: označava jedan nivo džoker znakova, kao što je a/+ koji odgovara a/k ili a/i.

#: označava više nivoa džoker znakova, kao što su a/# koji odgovaraju a/k, a/b/c/d.

## Kako MQTT radi?

### Quality of Service (QoS):

MKTT pruža tri vrste kvaliteta usluge i garantuje pouzdanost razmjene poruka u različitim mrežnim okruženjima.

QoS 0: Poruka se isporučuje najviše jednom. Ako klijent trenutno nije dostupan, izgubice ovu poruku.

QoS 1: Poruka se isporučuje najmanje jednom.

QoS 2: Poruka se isporučuje samo jednom.

# MQTT radni tok

Kada se razumiju osnovne komponente MKTT-a, može se pogledati kako funkcioniše opšti radni tok:

- ▶ Klijenti pokreću vezu sa brokerom koristeći TCP/IP, sa opcionim TLS/SSL enkripcijom za bezbjednu komunikaciju. Klijenti obezbjeđuju akreditive za autentifikaciju i navode čistu ili trajnu sesiju.
- ▶ Klijenti ili objavljaju poruke na određene teme ili se pretplate na teme da bi primali poruke. Klijenti izdavači šalju poruke brokeru, dok pretplatnici izražavaju interesovanje za primanje poruka o određenim temama.
- ▶ Broker prima objavljene poruke i prosleđuje ih svim klijentima koji su pretplaćeni na relevantne teme. Osigurava pouzdanu isporuku poruka prema specificiranom nivou kvaliteta usluge (QoS) i upravlja skladištenjem poruka za isključene klijente, na osnovu tipa sesije.

# Početak rada sa MQTT: Brzi vodič

Sada će se pokazati kako početi koristiti MQTT, sa nekoliko jednostavnih demonstracija.

Prije nego što se počne, potrebno je pripremiti MQTT brokera i MQTT klijenta.

## Priprema MQTT brokera

EMQX je skalabilna, distribuirana MQTT platforma za razmenu poruka koja podržava neograničen broj konekcija, nudi besprekornu integraciju i može da se primeni bilo gdje.

Obezbeđuje različite edicije koje zadovoljavaju različite zahtjeve korisnika.

# Početak rada sa MQTT: Brzi vodič

## Priprema MQTT brokera

EMQX Serverless je MQTT usluga za više zakupaca s cijenama prema korištenju i funkcijama automatskog skaliranja. Može se pokrenuti za nekoliko minuta i radi u 17 regija na AWS-u, Google Cloudu i Microsoft Azureu.

Isprobajte EMQX Serverless. Zauvijek besplatno za manje od 1 miliona minuta sesije mjesečno. Besplatni javni MQTT broker.

U ovom vodiču koristit ćemo se besplatnim javnim MQTT brokerom koji nudi EMQ, izgrađen na EMQX platformi.

Pojedinosti o pristupu serveru su sljedeće:

Adresa brokera: [broker.emqx.io](http://broker.emqx.io)

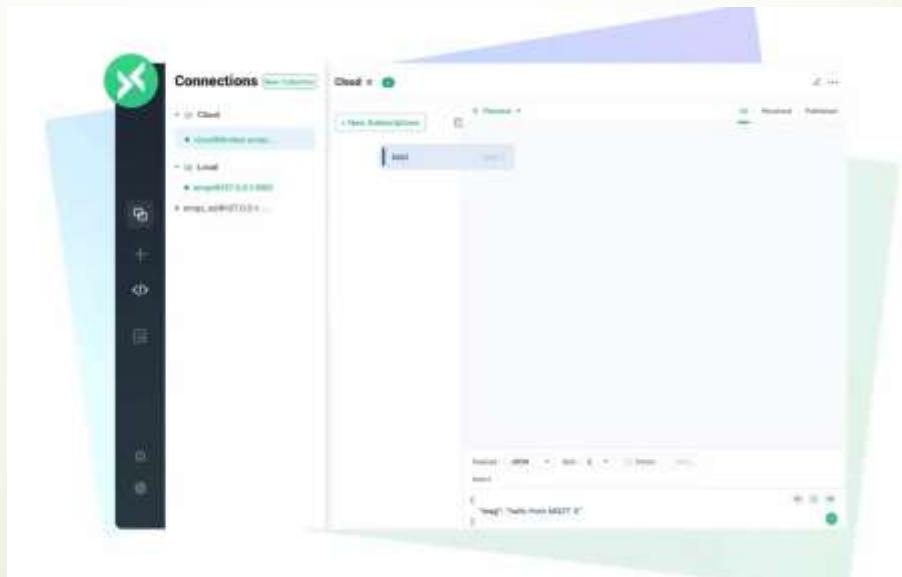
TCP priključak: 1883

WebSocket priključak: 8083

# Početak rada sa MQTT: Brzi vodič

## Priprema MQTT klijenta

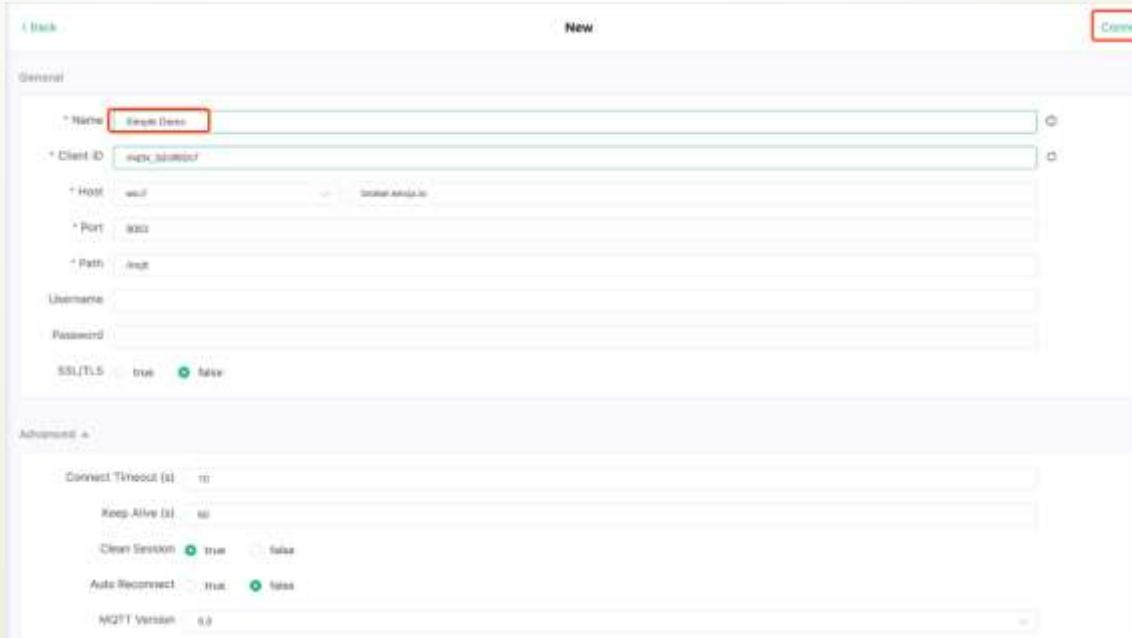
- ▶ Ovom prilikom će se koristiti MQTT klijentski alat koji nudi **MQTTX** i koji podržava pristup putem linka: <http://www.emqx.io/online-mqtt-client>. MQTT X takođe nudi **desktop client** i **command line tool**.
- ▶ **MQTTX** je elegantan višeplatformski **MQTT 5.0** desktop klijent koji radi na macOS, Linux i Windows. Njegov interfejs u stilu časkanja omogućava korisnicima da lako kreiraju više MQTT veza i pretplata za MQTT poruke.



# Početak rada sa MQTT: Brzi vodič

## Kreiranje MQTT konekcije

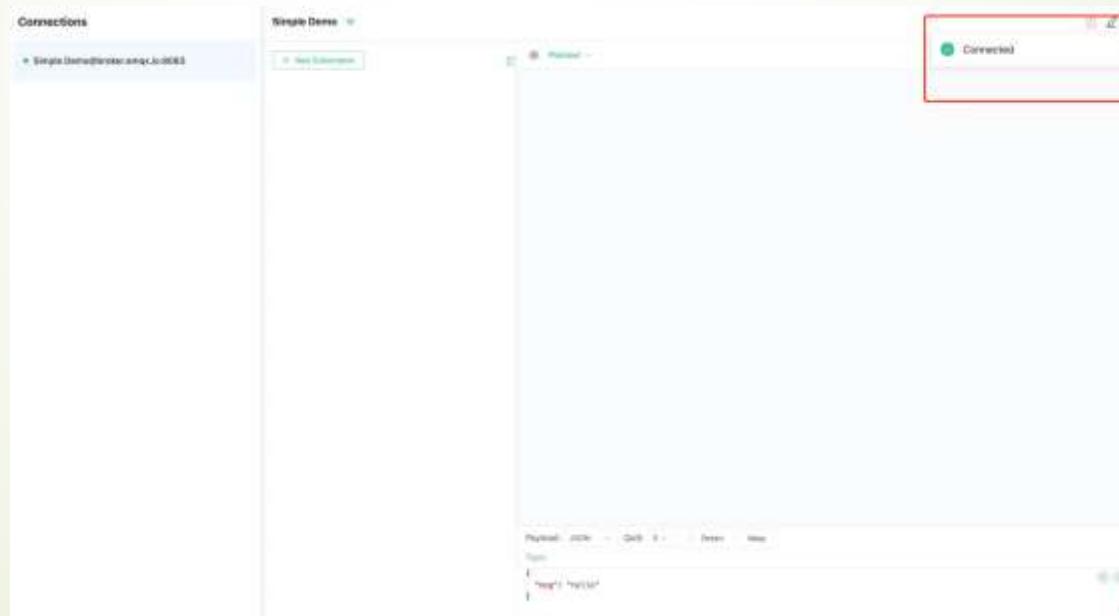
- ▶ Prije korištenja MQTT protokola za komunikaciju, klijent mora kreirati MQTT vezu za povezivanje s brokerom.
- ▶ Treba ići na: <http://www.emqx.io/online-mqtt-client> i kliknite na dugme New Connection na sredini stranice i vidjet ćete sljedeću stranicu.



# Početak rada sa MQTT: Brzi vodič

## Kreiranje MQTT konekcije

- Unesite **Proba** u Name i kliknite dugme Connect u gornjem desnom uglu za stvaranje MQTT veze. Sljedeće označava da je veza uspješno uspostavljena.

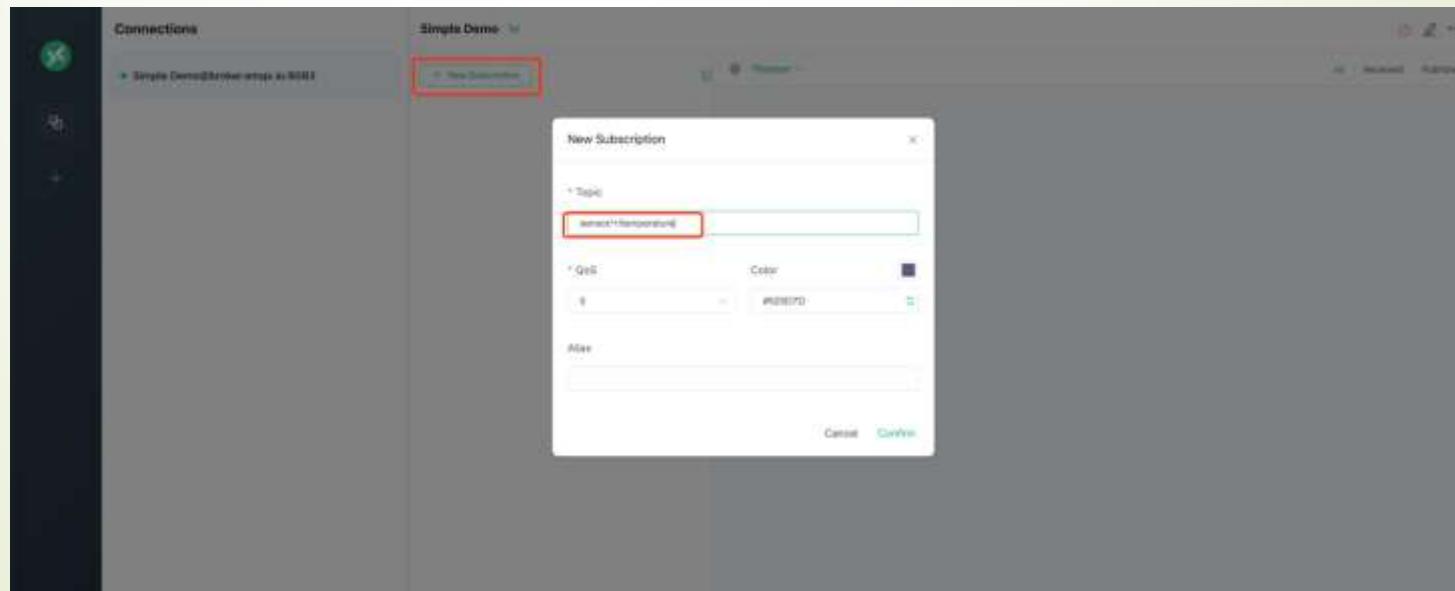


Kako biste saznali više o parametrima MQTT veze, pogledajte post na blogu:  
[How to Set Parameters When Establishing an MQTT Connection](#).

# Početak rada sa MQTT: Brzi vodič

Pretplata se na temu:

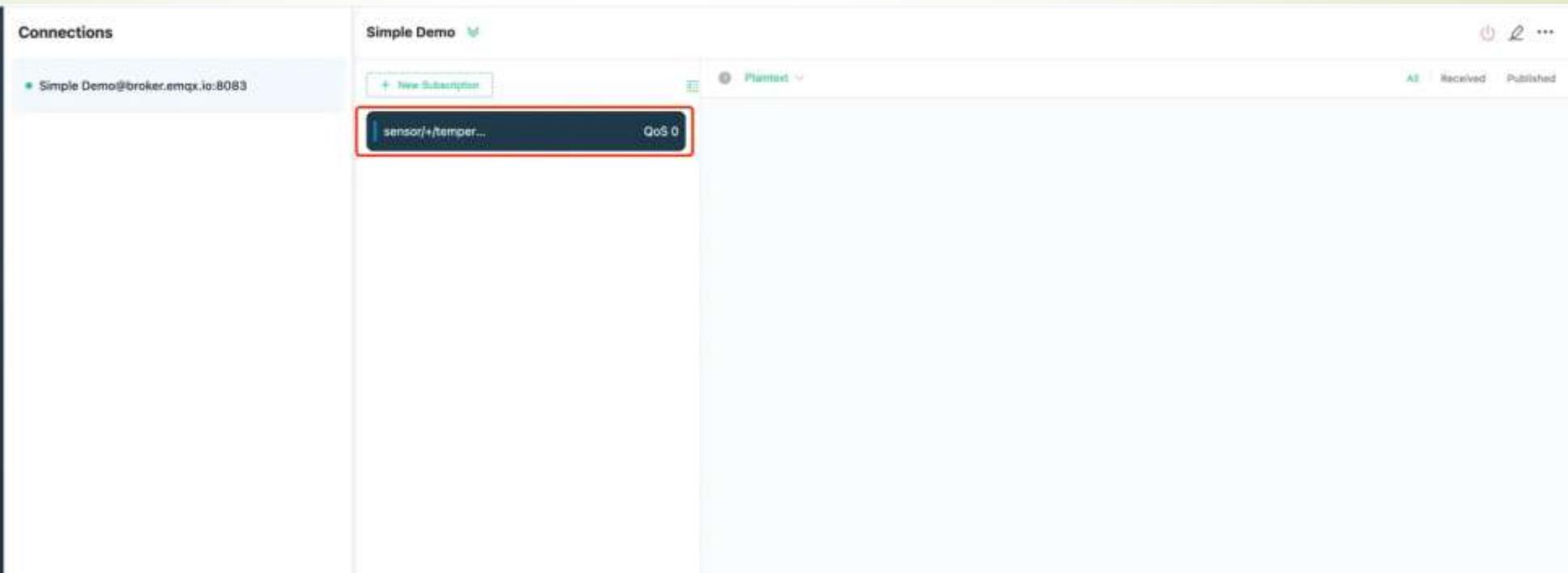
- ▶ Pretplaćivanje na temu sa zamjenskim znakom **senzor+/temperatura** u Proba vezi stvorenoj ranije, koja će primati podatke o temperaturi koje objavljaju svi senzori.
- ▶ Kao što je prikazano u nastavku, kliknite na dugme New Subscription i unesite temu **senzor+/temperatura** u polje Topic u iskačućem okviru, zadržavajući zadani QoS na 0.



# Početak rada sa MQTT: Brzi vodič

Pretplata se na temu:

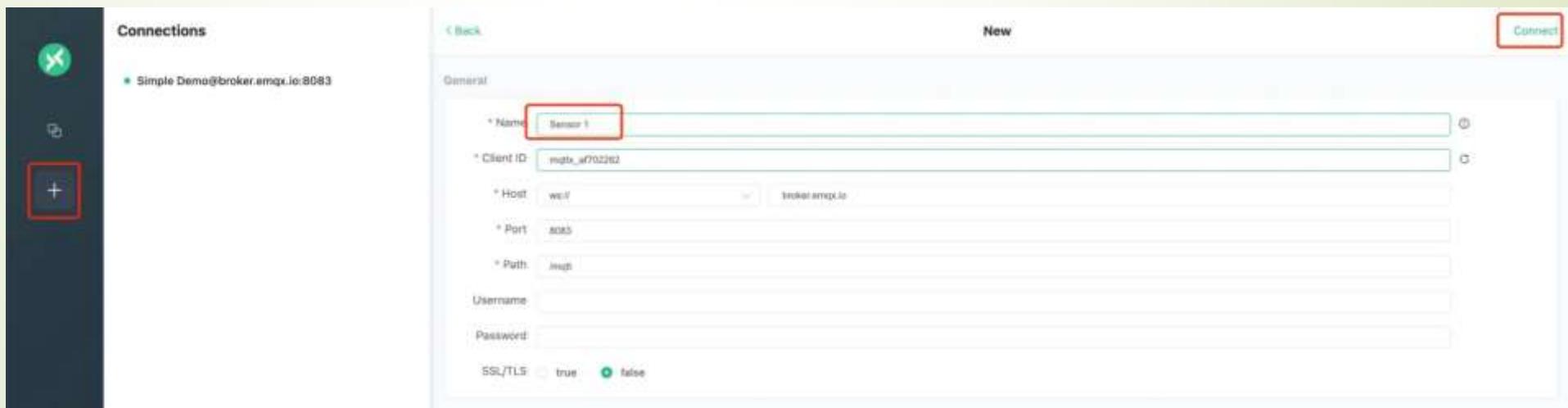
- ▶ Nakon uspješnog pretplaćivanja, vidjet ćete dodatni zapis u sredini liste pretplata.



# Početak rada sa MQTT: Brzi vodič

## Objavljivanje MQTT poruka:

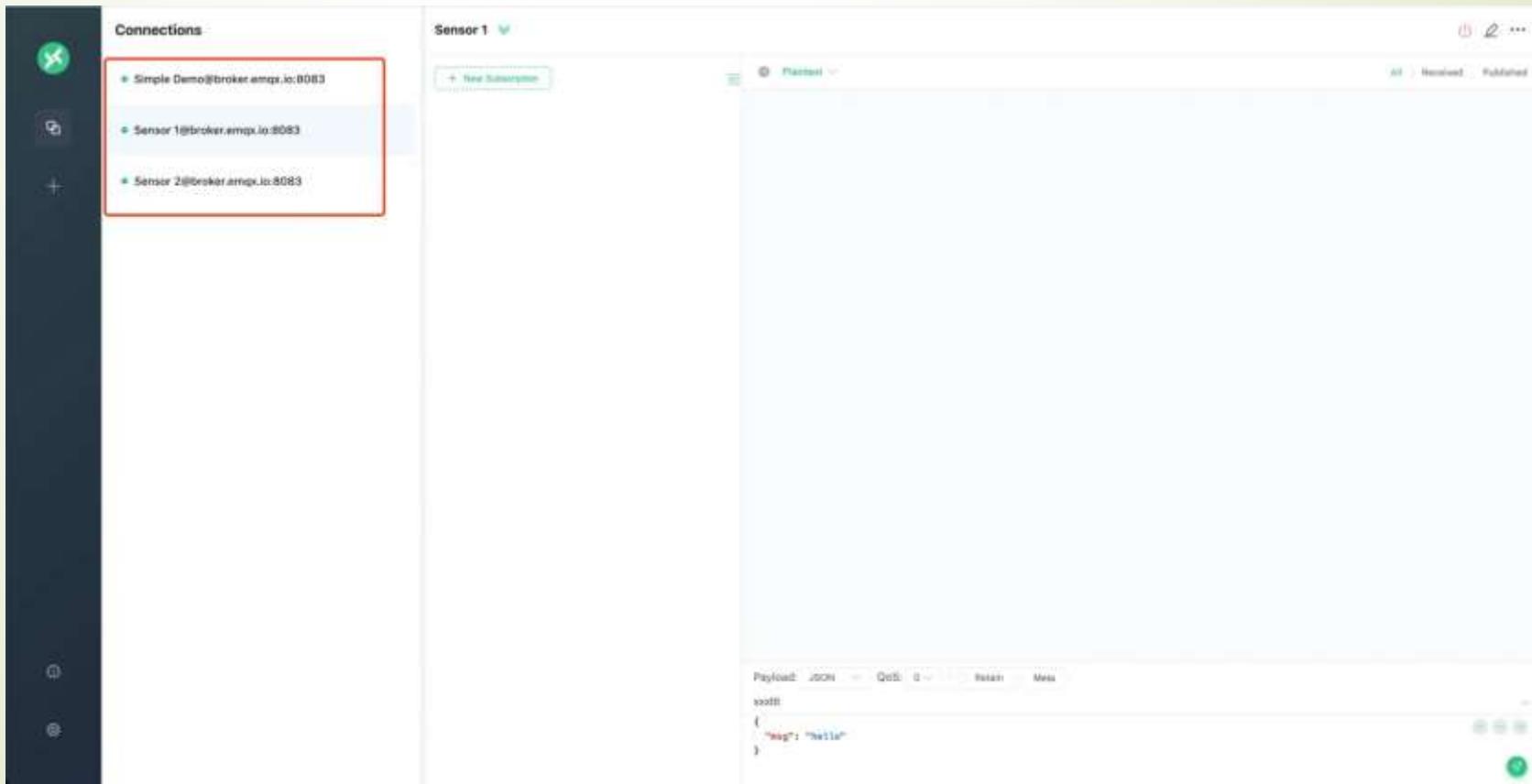
- ▶ Zatim treba kliknuti dugme + na lijevom meniju kako bismo stvorili dvije veze, senzor 1 i senzor 2, kako bismo simulirali dva temperaturna senzora.



# Početak rada sa MQTT: Brzi vodič

## Objavljivanje MQTT poruka:

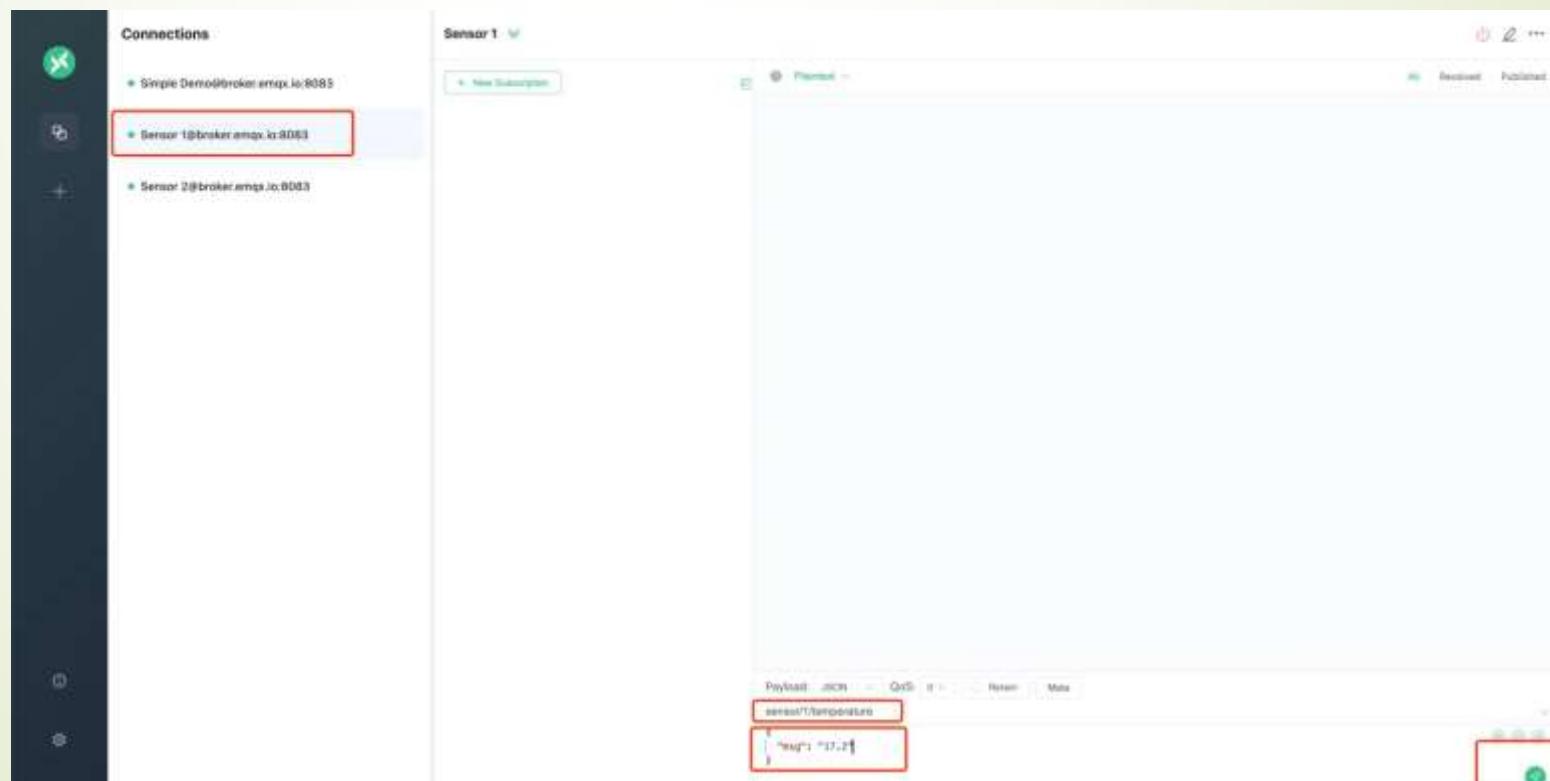
- ▶ Nakon što se veza kreira, vidjet ćete tri veze, a sve tačkice mrežnog statusa s lijeve strane veza bit će zelene.



# Početak rada sa MQTT: Brzi vodič

## Objavljivanje MQTT poruka:

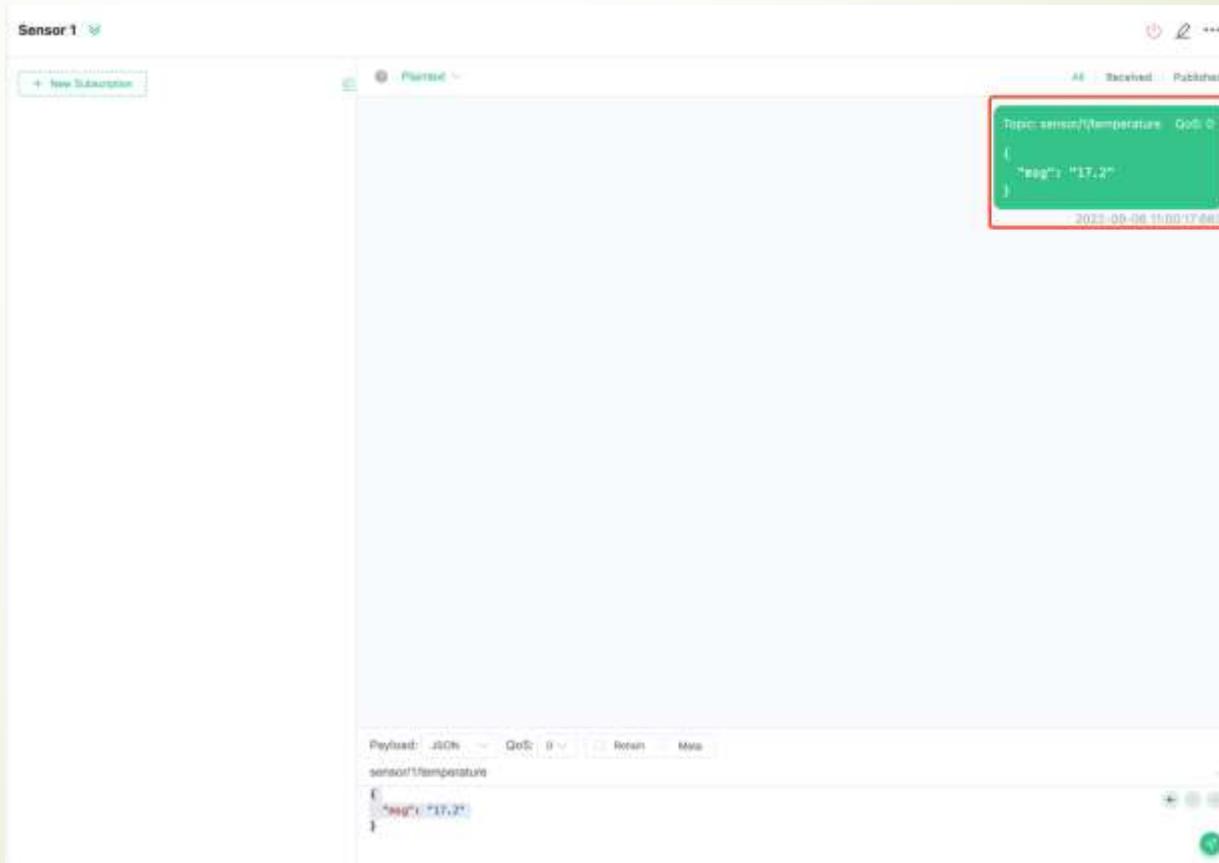
- Izaborite vezu **Senzor 1**, i unesite temu za objavljivanje **senzor/1/temperatura** u donjem levom delu stranice, unesite sledeću poruku, u formatu JSON, u okvir za poruke i kliknite na dugme za objavljivanje u donjem desnom uglu da biste poslali poruku.



# Početak rada sa MQTT: Brzi vodič

## Objavljivanje MQTT poruka:

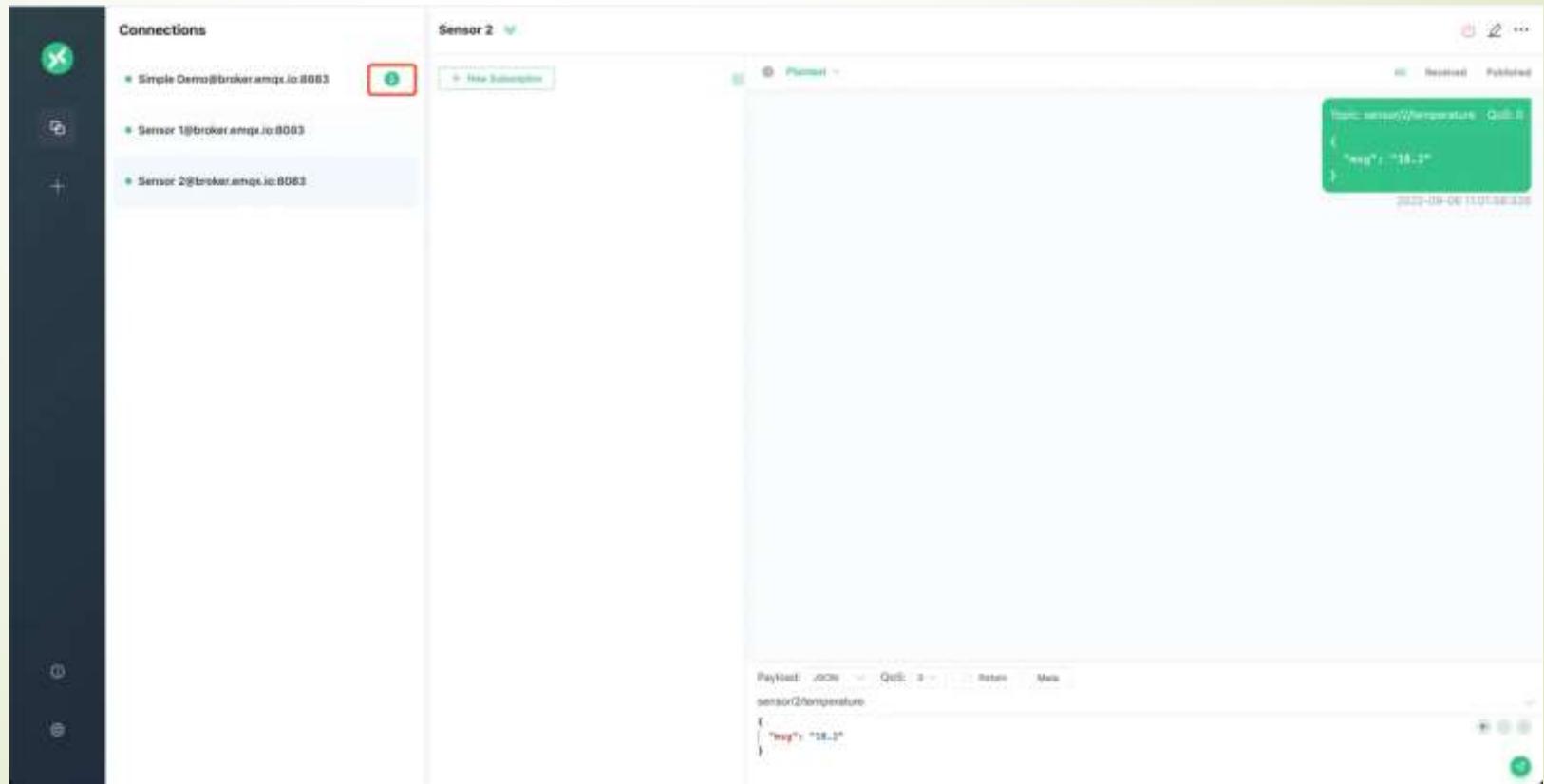
- ▶ Poruka je uspješno poslana na sljedeći način.



# Početak rada sa MQTT: Brzi vodič

## Objavljivanje MQTT poruka:

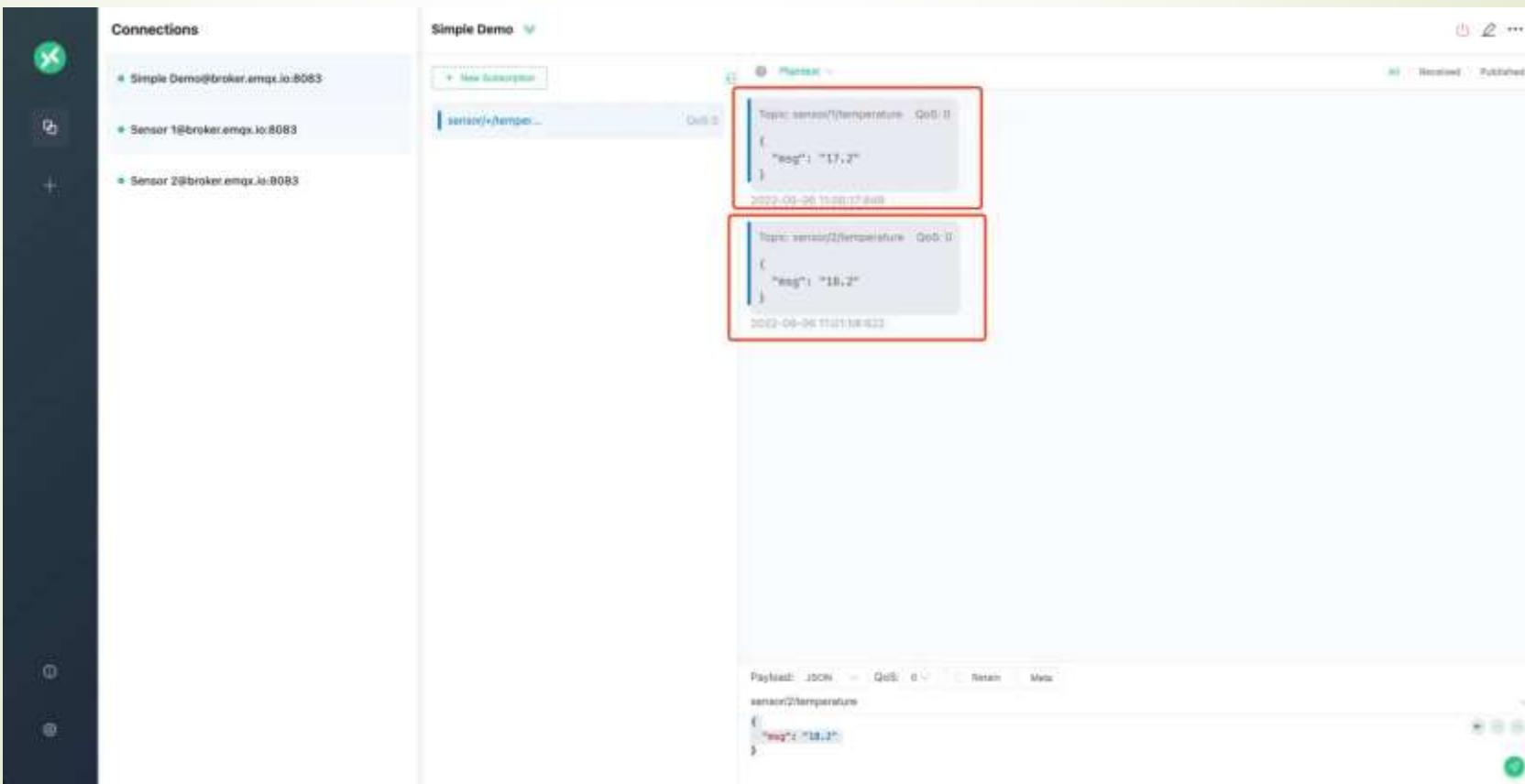
- ▶ Koristeći iste korake, objavite sljedeću JSON poruku u temi **senzor/2/temperatura** u vezi senzora 2.



# Početak rada sa MQTT: Brzi vodič

## Objavljivanje MQTT poruka:

- ▶ Kliknite vezu **Proba** i vidjet ćete dvije poruke koje su poslala dva senzora.



# Arduino uređaj kao klijent

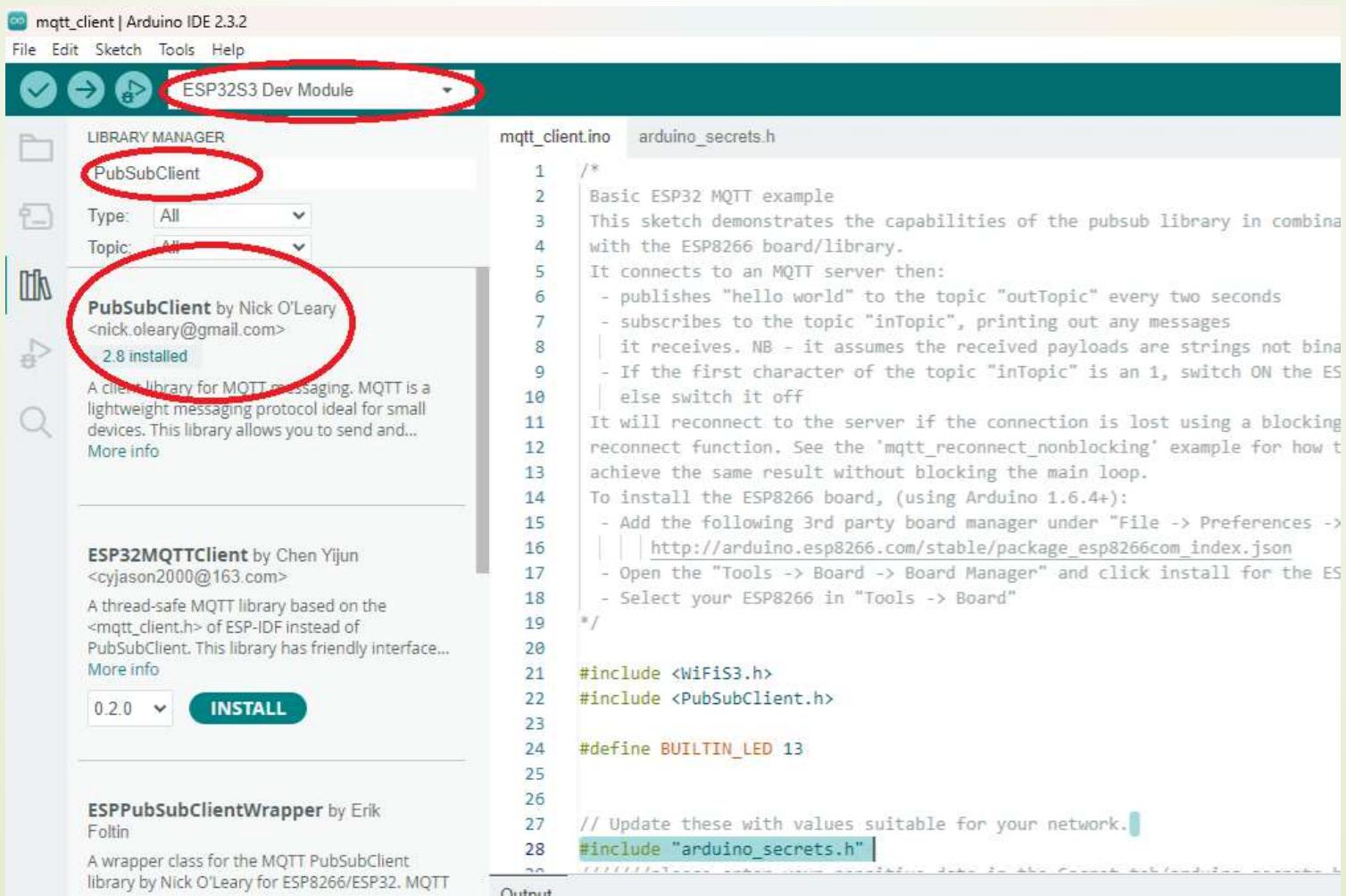
Upišite u Arduino Uno R4 WiFi skeč nazvan:  
**mqtt\_cilent.ino**

Skeč omogućuje da adruino uređaj funkcioniše kao MQTT klijent.

Da bi se skeč uspješno kompajlirao i izvršavao potrebno je instalirati biblioteku: **PubSubClient**

# Arduino uređaj kao klijent

Instaliranje PubSubClient biblioteke:



# Arduino uređaj kao klijent – uloga `loop()` funkcije

U biblioteci `PubSubClient`, funkcija `client.loop()` ključna je za održavanje veze s **MQTT brokerom** i rukovanje dolaznim porukama.

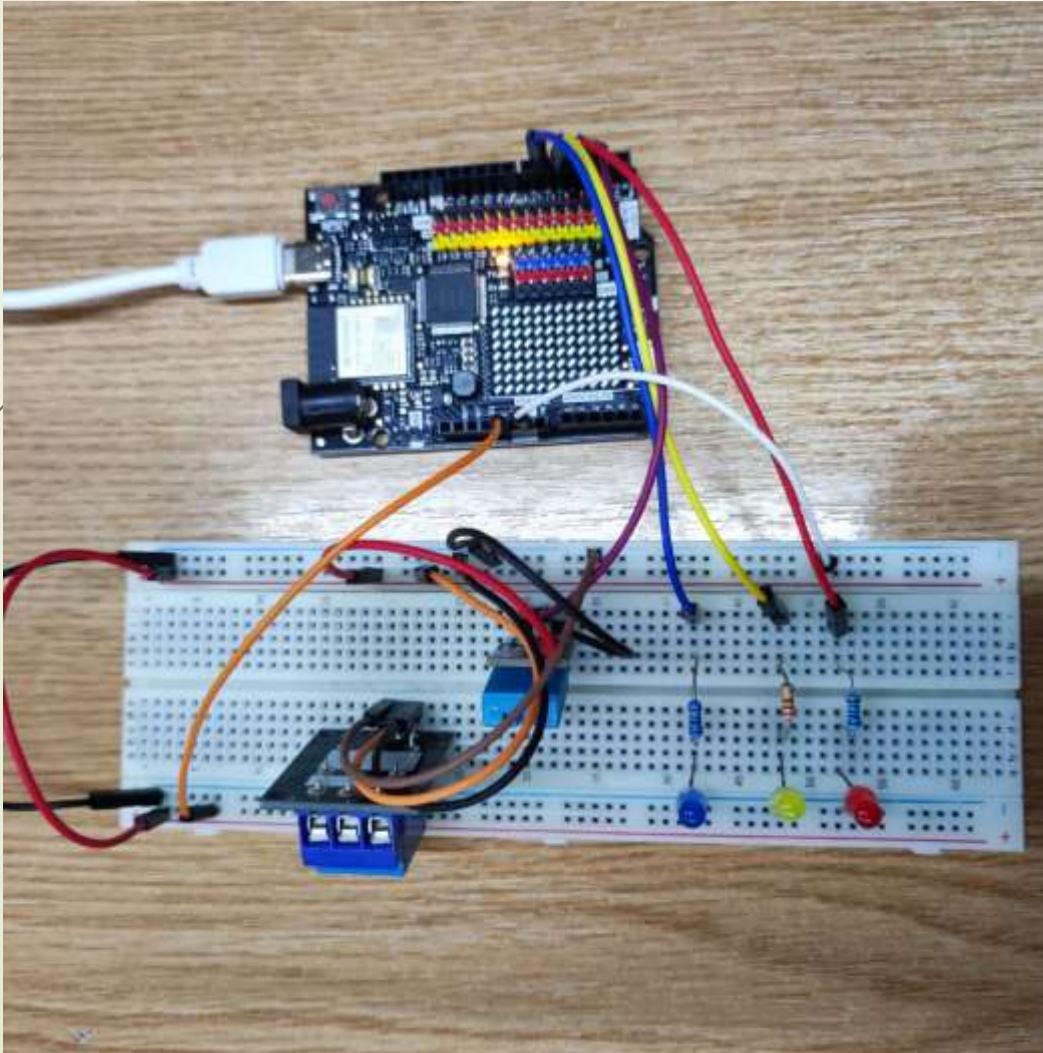
Kada se pretplatite na teme ili objavite poruke koristeći MQTT (Message Queuing Telemetry Transport), mikrokontrolerska ploča mora stalno komunicirati s MQTT brokerom kako bi primala poruke ili slala ažuriranja. Za ovu kontinuiranu komunikaciju odgovorna je funkcija `client.loop()`.

Evo što radi:

- **Održavanje veze:** osigurava da se veza s MQTT brokerom održava živom. MQTT se temelji na trajnoj TCP vezi, a klijent mora slati periodične kontrolne pakete kako bi održao ovu vezu.
- **Rukovanje dolaznim porukama:** Provjerava sve dolazne poruke iz pretplaćenih tema. Kada stignu poruke, funkcija `loop()` ih obrađuje i pokreće odgovarajuće funkcije povratnog poziva koje ste možda definisali u svom kodu.
- **Obrada odlaznih poruka:** Ako postoje poruke u redu čekanja za objavljivanje, funkcija `loop()` se brine za njihovo slanje MQTT brokeru.
- **Keep-Alive Mechanism:** Takođe upravlja mehanizmom za održavanje aktivnosti MQTT protokola, gdje povremeno šalje ping zahtjeve brokeru kako bi se osiguralo da je veza još uvijek aktivna.

Sve u svemu, redovno pozivanje `client.loop()`, obično unutar glavne petlje vašeg programa, bitno je za nesmetan rad MQTT komunikacije. Održava komunikaciju živom, obrađuje dolazne poruke i osigurava da vaš uređaj i dalje reaguje na MQTT mrežu.

# Priprema za vježbu



## Mikrokontrolerski dizajn

Crvena LED je preko otpornika  $220\Omega$  spojena na pin 6.

Žuta LED je preko otpornika  $220\Omega$  spojena na pin 5.

Plava LED je preko otpornika  $220\Omega$  spojena na pin 3.

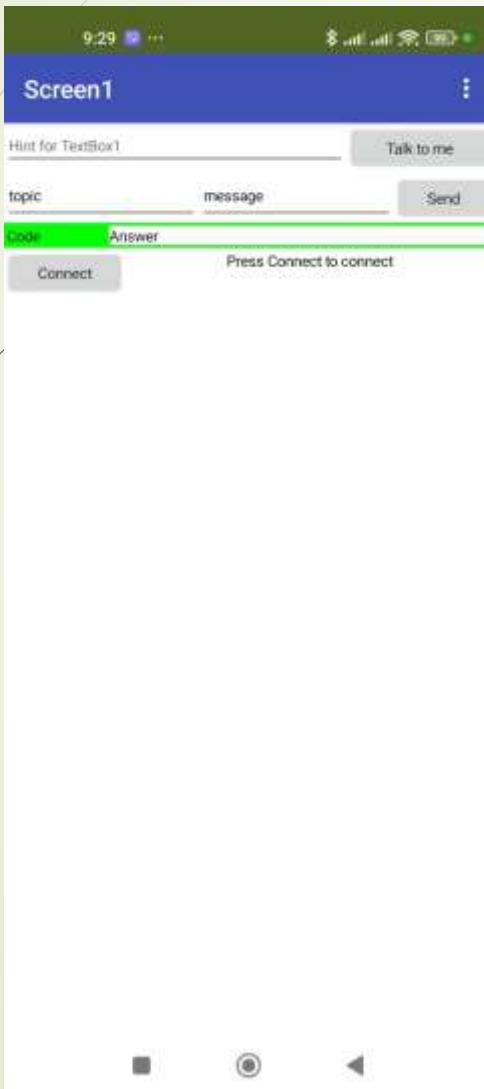
Signal Relay-a je spojen na pin A0. Uz to VCC je spojeno na 5V, a GND na GND.

Signal DHT11 senzora povezan je na pin 2. VCC je spojeno na 5V, a GND na GND

Link za preuzimanje MQTT extenzije:

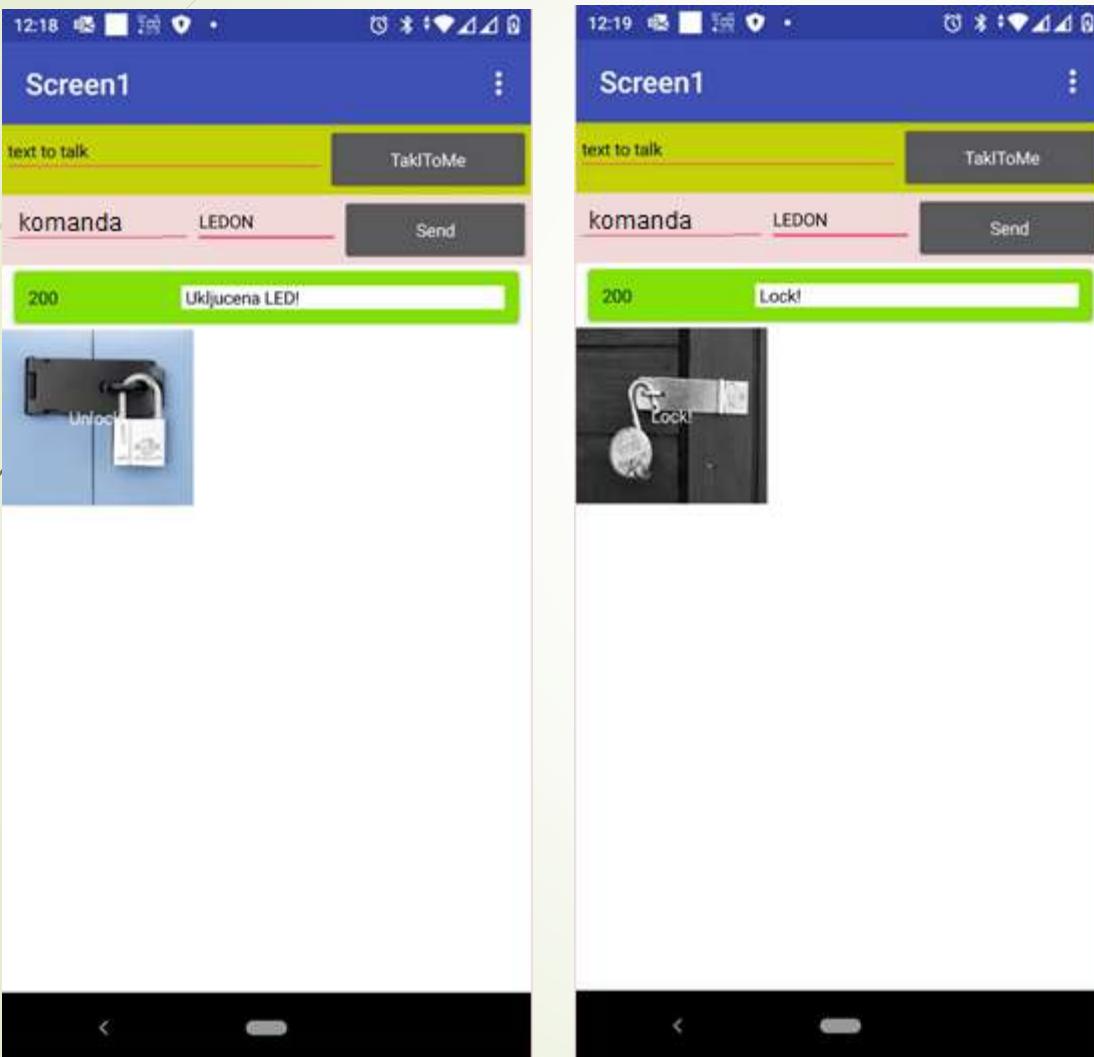
<https://ullisroboterseite.de/android-AI2-PahoMQTT-en.html#down>

# Priprema za vježbu



Modifikovati ćemo aplikaciju iz predhodne nastave, tako da sada podržava povezivanje putem MQTT protokola.

# Vježba 1



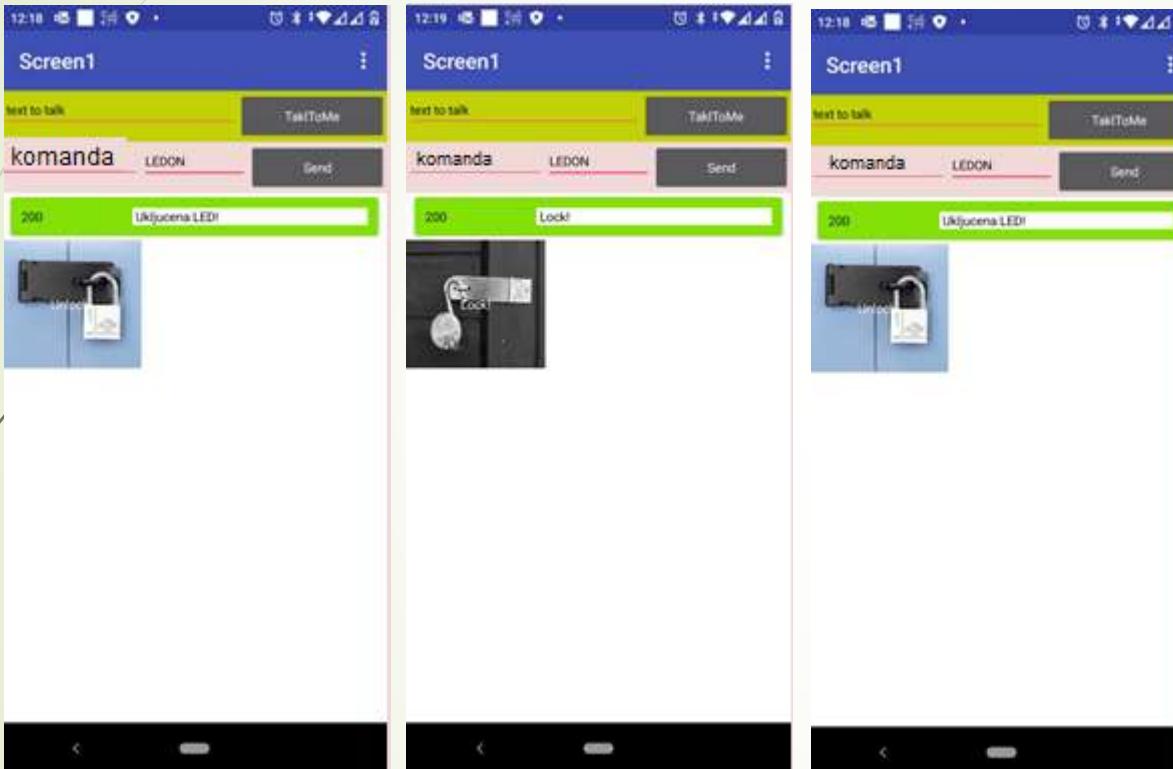
Dopuniti aplikaciju mogućnošću otvaranja brave (odnosno upravljanja rele-om).

Kada se pritisne komandno dugme, sa slikom zaključanog katanca, zatvara se NO kontakt rele-a u mikrokontroleskom dizajnu, a slika na komandnom dugmetu mobilnog telefona se mijenje u sliku sa otvorenim katancem.

Kada se pritisne komandno dugme, sa slikom otključanog katanca, otvara se NO kontakt rele-a u mikrokontroleskom dizajnu, a slika na komandnom dugmetu mobilnog telefona se mijenje u sliku sa zatvorenim katancem.

(1 – 0.5 bodova)

# Vježba 2

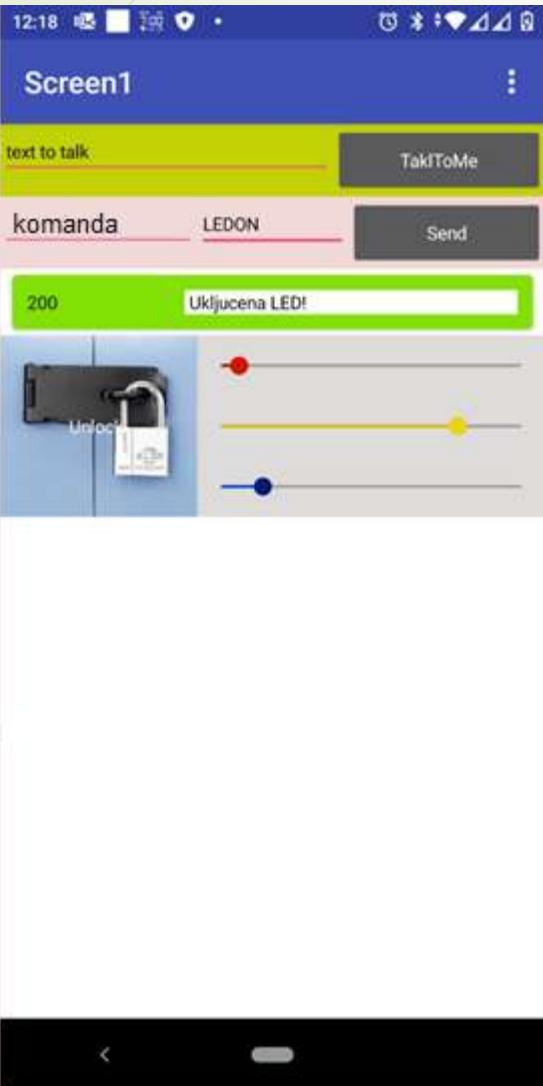


Dopuniti rad sa rele-om, na sljedeći način:

Ukoliko se po zatvaranju kontakta rele-a, u roku od 5 sekundi ne zada komanda za otvaranje kontakta, kontakt se otvara, a komandno dugme ponovo dobija sliku zaključanog katanca.

(1 – 0.5 bodova)

# Vježba 3



Dopuniti aplikaciju mogućnošću upravljanja intezitetom sjaja crvene, žute i plave LED, koje su povezane na pinove 3,5 i 6.

Upotrijebiti *Slider* komponente.

(2 – 1 – 0.5 bodova)

# Vježba 4



Dopuniti aplikaciju mogućnošću mogućnošću prikazivanja, informacija o temperaturi i vlazi.

Informaciju o temperaturi i vlazi obezbeđuje senzor temperature i vlage u mikrokontrolerskom dizajnu.

Informacije prikazati kao brojčani ispis na komandnom dugmetu, sa odgovarajućom sugestivnom slikom u pozadini, kao i pomoću slider komponenti.

Informacije obezbijediti bez pritiska na komandno dugme i bez bilo kakve manuelne akcije na telefonu

(2 – 1 - 0.5 bodova)

U ovom primjeru koisti se **Adafruit DHT Humidity & Temperature Sensor** biblioteka koja se može preuzeti sa [ovog linka \(GitHub\)](#).

Biblioteka DHT senzora radi jedino ako se ima instalirana **Adafruit\_Sensor** biblioteka, ona se može preuzeti sa [ovog linka \(GitHub\)](#).

# Vježba 5



Dopuniti aplikaciju sljedećim:

Po pritisku komandnog dugmeta za prikaz temperature, istu prikazati u Farenheit-ima, u trajanju 5 sekundi.

(1 – 0.5 bodova)

U ovom primjeru koisti se **Adafruit DHT Humidity & Temperature Sensor** biblioteka koja se može preuzeti sa [ovog linka \(GitHub\)](#).

Biblioteka DHT senzora radi jedino ako se ima instalirana **Adafruit\_Sensor** biblioteka, ona se može preuzeti sa [ovog linka \(GitHub\)](#).

# Vježba 6



Upravljati LED glasovnim komandama:

Za uključenje LED: "turn on red(blue/yellow)"

Za isključenje LED: "turn off red(blue/yellow)"

Prilikom uključenja LED, ista treba sijati  
intezitetom zadanim pomoću slajdera

(1 – 0.5 bodova)

# Vježba 7



Upravljati koračnim motorom glasovnim komandama:

LEFT, RIGHT, FASTER, SLOWER, STOP

Uloga glasovnih dugmadi je jasna iz samog naziva.

Na ekranu mobilnog telefona prikazivati brzinu (kor/s) i smjer okretanja koračnog motora.

(2 – 1 – 0.5 bodova)

# Vježba 8



Kada se mobilni telefon primkne uhu, na LCD-u uređaja, u njegovom gornjem redu, ispisati poruku: „Telefon pri uhu!“.

Kada se telefon odmakne od uha, na LCD-u uređaja, u njegovom gornjem redu, ispisati poruku: „Telefon odmaknut“.

(1 – 0.5 bodova)

U ovom primjeru koristi se I<sup>2</sup>C LCD. Biblioteka za rad sa njime može se preuzeti sa linka:

[https://github.com/johnrickman/LiquidCrystal\\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C)