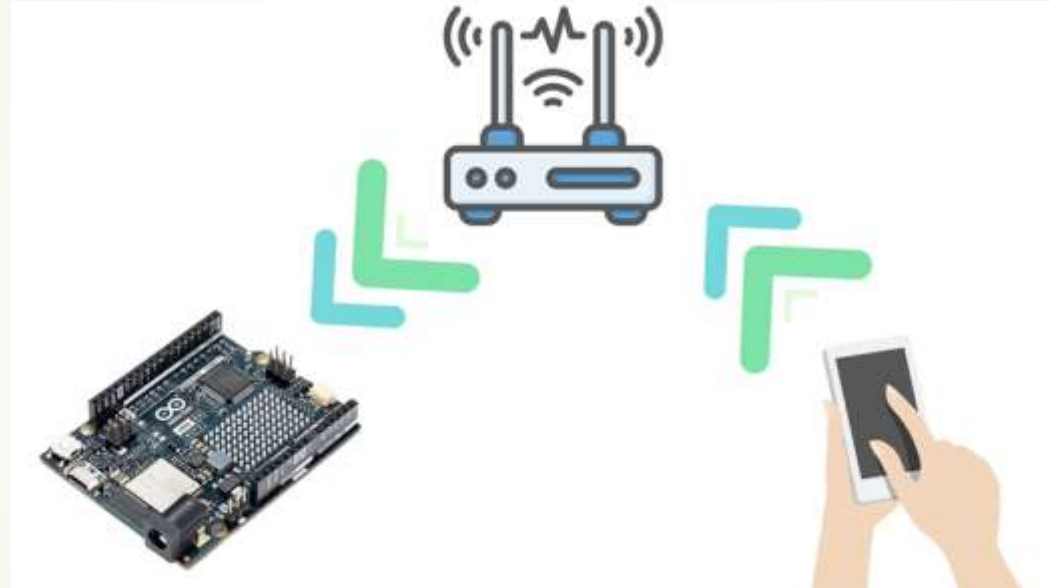


Upravljanje uređajem pomoću mobilnog telefona (MQTT protokol)

Komuniciranje sa Arduino R4 WiFi bezžično posredstvom WiFi AP(Access Point)-a

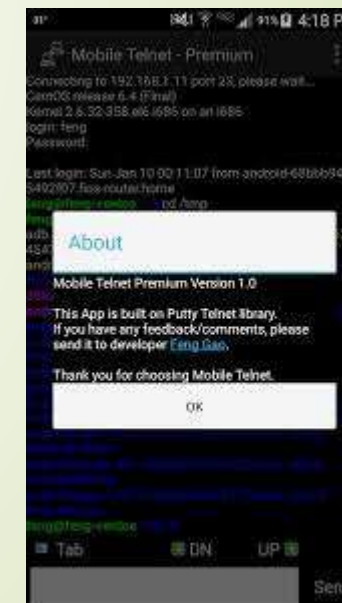
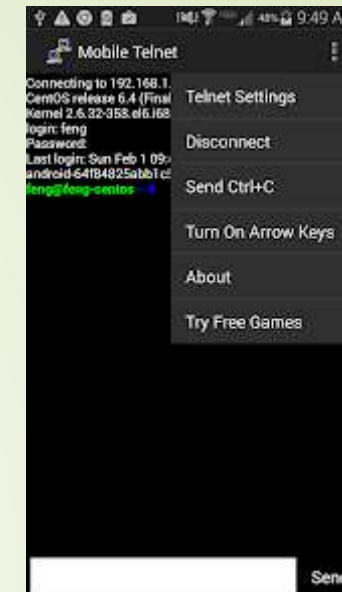


- Arduino će raditi u AP (Access Point) + Station (STA) modu. Što znači da će drugim uređajima omogućavati pristup putem WiFi mreže.
- Proces je jednostavan. Upotrijebite telefon da pošaljete bilo koju komandu Arduino. Uz pomoć WiFi modula u Arduino R4 WiFi ploči sve će raditi bezžično.

Potrebni elementi

Upišite **AP2024.ino** skeč u Arduino

- Podešavanje telefona:
 - Instalirajte TCP klinta za Android
 - Možete instalirati bilo koji TCP klijent koji je raspoloživ na Play Store, a na Android telefonu možete i baš Mobile Telnet
 - Povežite vaš sa Arduino R4 WiFi uređajem
 - Nakon povezivanja otvorite Mobile Telnet aplikaciju
 - Kreirajte konekciju klikom na „Telnet Settings“ opciju menija i dodajte IP (Host Name) i Port
 - Povežite se klikom na “Connect” opciju menija





Već podržane poruke

- Kažite: HELLO
- Kažite: Kako ide?
- Uključite LED komandom LEDON
- Ili isključite LED komandom LEDOFF
- Možete mijenjati i dalje nadograđivati.



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 pretplati.

Dizajniran je za uređaje sa ograničenim resursima i niske propusnosti, velikih kašnjenja ili nepouzdana 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 ekosistemima 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 pretplate 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-pretplata 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, podstičući besprekornu komunikaciju i interoperabilnost u različitim IoT ekosistemima. MQTT se može koristiti u C++-u, PHP-u, Node.js, Python, Golang, Node.js 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-pretplate,
- 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 vezom, prekidom, zahtevima za pretplatu i otkazivanje pretplate i porukama za rutiranje.

Moćan MQTT broker može da podrži brojne veze i protok poruka na millionskom 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?

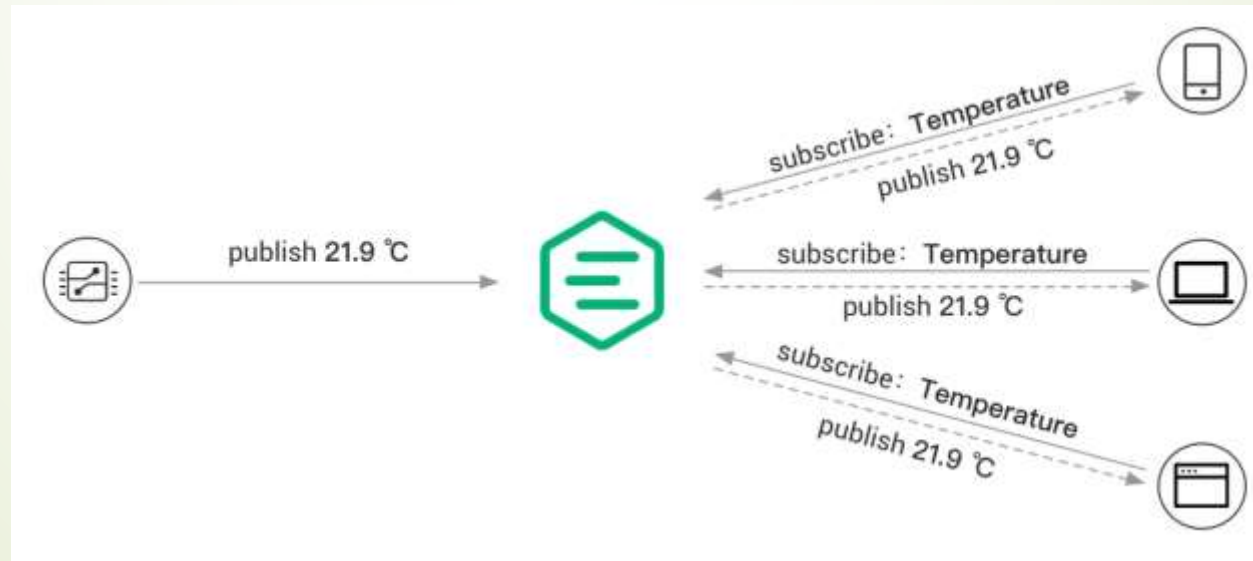
Obrazac objava-pretplata:

Obrazac objava-pretplata 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-pretplata:

Dijagram prikazuje MQTT proces objavljivanja/pretplata. 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 pretplaćen 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

MQTT 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):

MQTT 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 MQTT-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 objavljuju 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.

Obezbjeđuje 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`

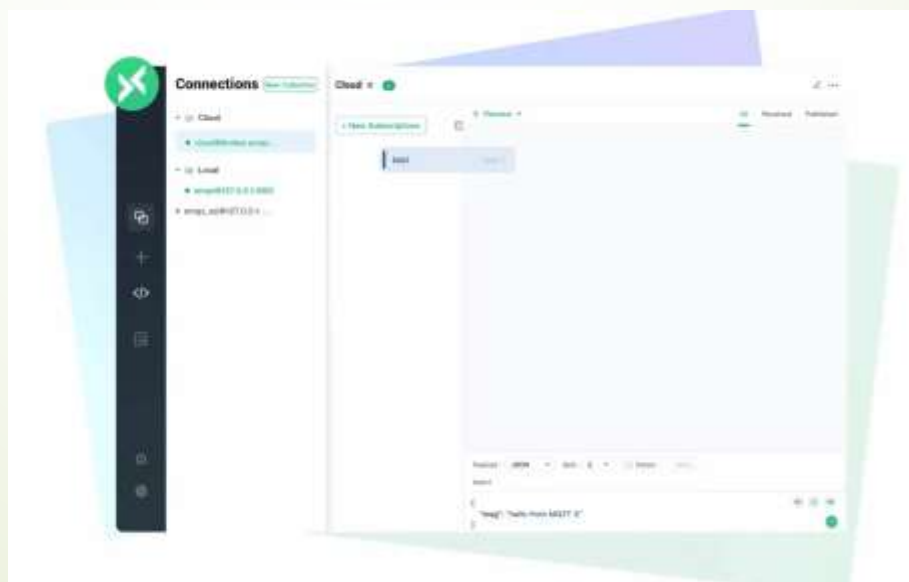
TCP priključak: 1883

WebSocket priključak: 8083

Početak rada sa **MQTT**: Brzi vodič

Priprema MQTT klijenta

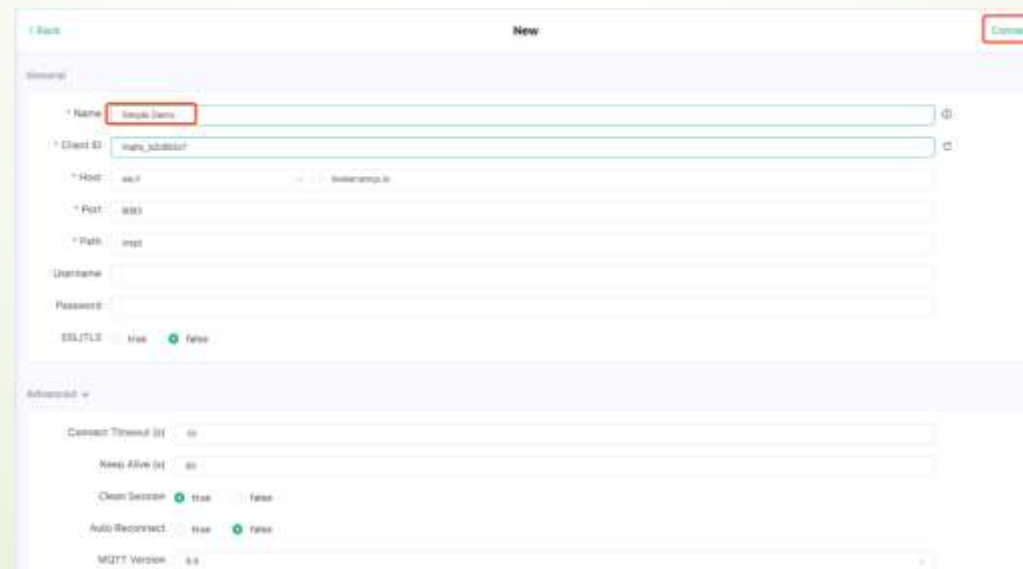
- Ovom prilikom će se koristiti MQTT klijentski alat koji nudi **MQTTX** i koji podržava pristup putem linka: <http://www.emqx.io/online-mqtt-client>. MQTT X takođe nudi **desktop client** i **command line tool**.
- **MQTTX** je elegantan višepatformski **MQTT 5.0** desktop klijent koji radi na macOS,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

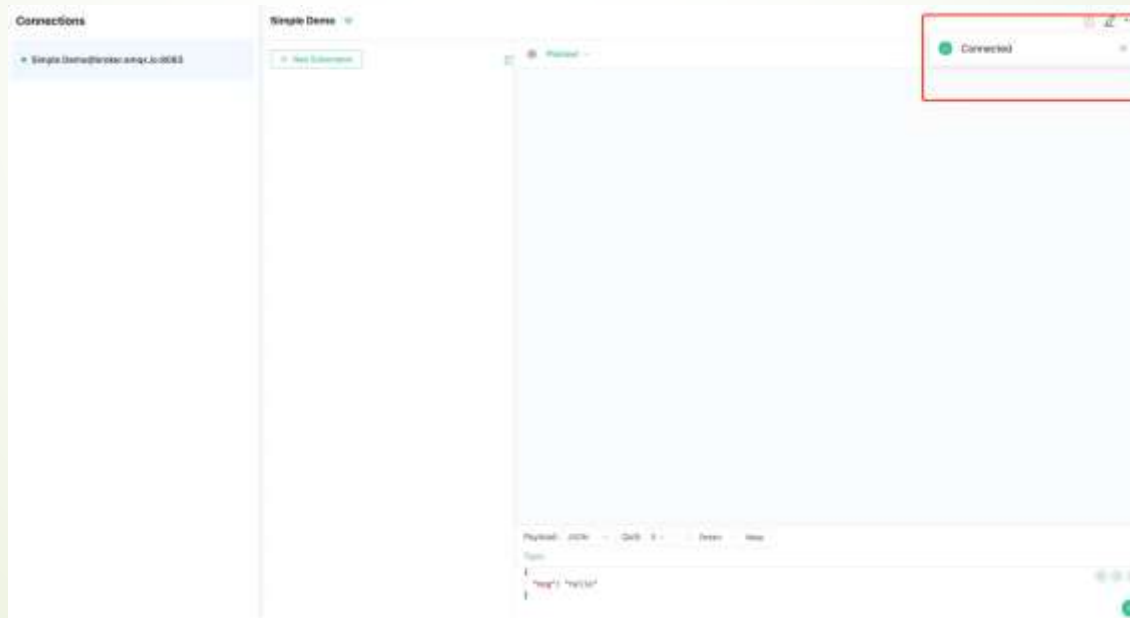


The screenshot displays the 'New' connection configuration page of an MQTT client. The interface is divided into 'General' and 'Advanced' sections. In the 'General' section, the 'Name' field is highlighted with a red box and contains the text 'mqtt-demo'. Other fields include 'Client ID' (mqtt_123456789), 'Host' (mqtt.emqx.io), 'Port' (1883), and 'Path' (/mqtt). There are also fields for 'Username' and 'Password'. The 'SSL/TLS' option is set to 'false'. The 'Advanced' section includes 'Connect Timeout' (10s), 'Keep Alive' (60s), 'Clean Session' (checked), 'Auto-Reconnect' (checked), and 'MQTT Version' (3.1).

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.

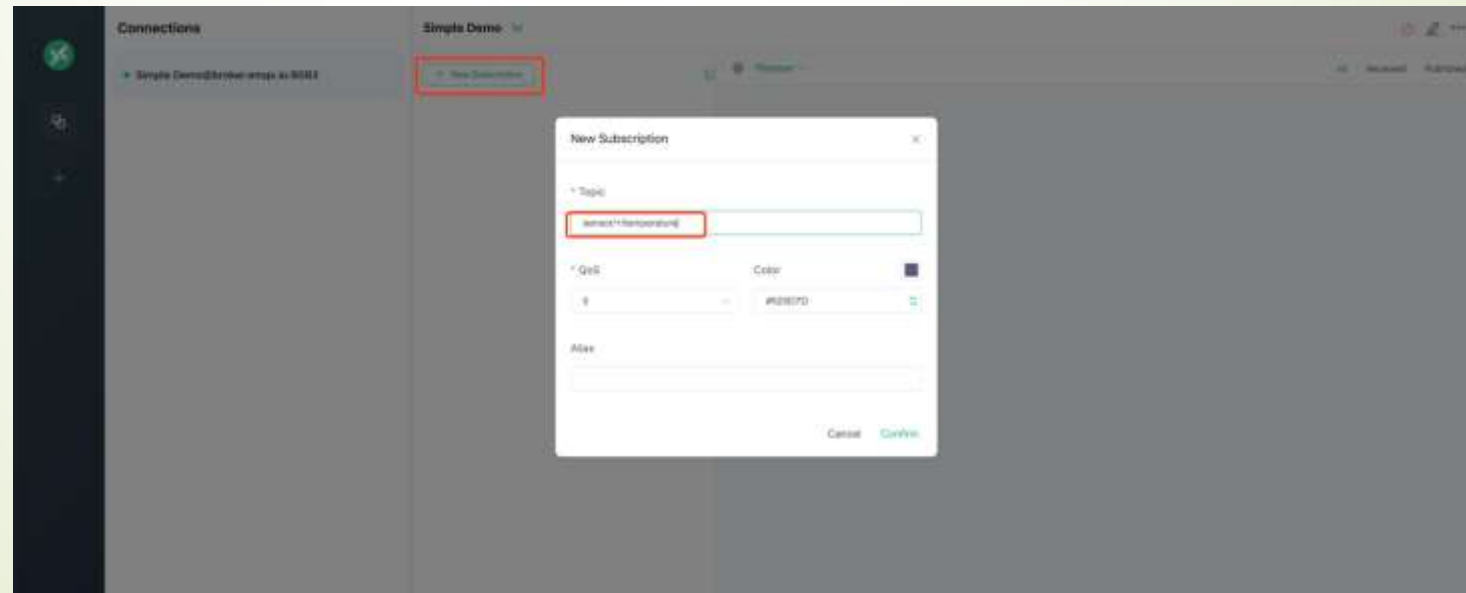


Kako biste saznali više o parametrima MQTT veze, pogledajte post na blogu: [How to Set Parameters When Establishing an MQTT Connection](#).

Početak rada sa **MQTT**: Brzi vodič

Pretplata se na temu:

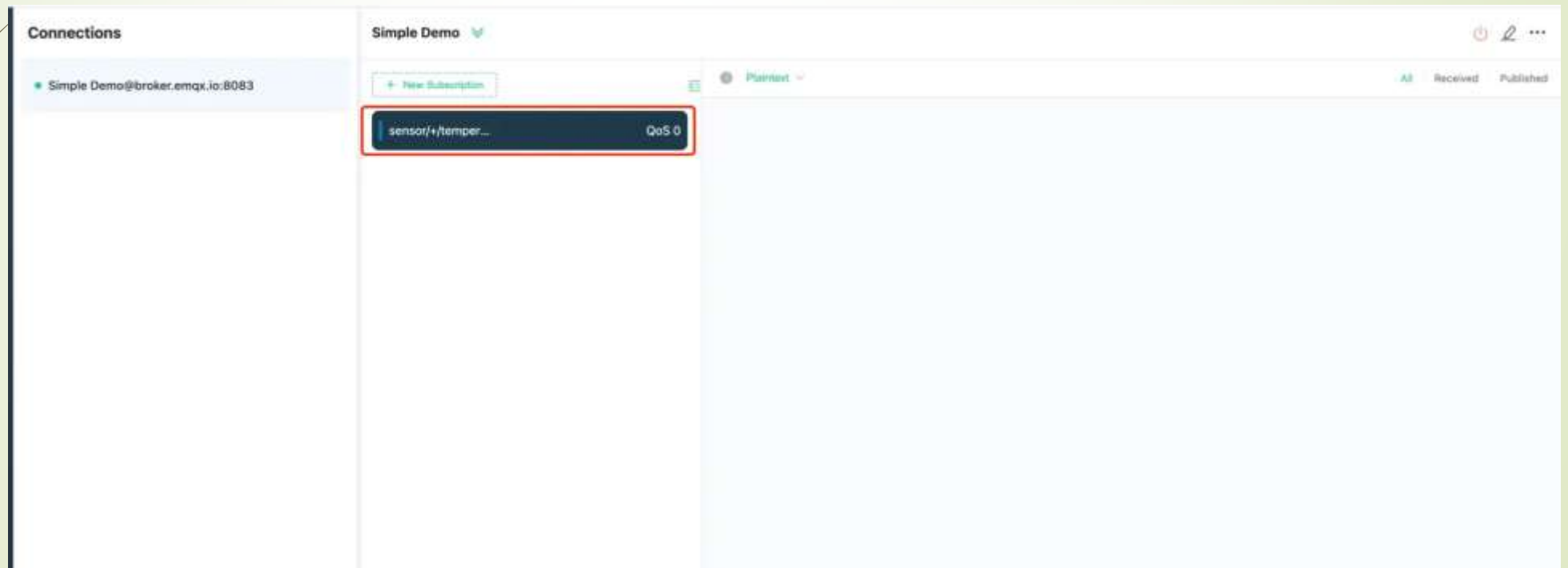
- Pretplaćivanje na temu sa zamjenskim znakom **senzor+/temperatura** u **Proba** vezi stvorenoj ranije, koja će primiti podatke o temperaturi koje objavljuju svi senzori.
- Kao što je prikazano u nastavku, kliknite na dugme **NEW SUBSCRIPTION** i unesite temu **senzor+/temperatura** u polje **TOPIC** u iskačućem okviru, zadržavajući zadani QoS na 0.



Početak rada sa **MQTT**: Brzi vodič

Pretplata se na temu:

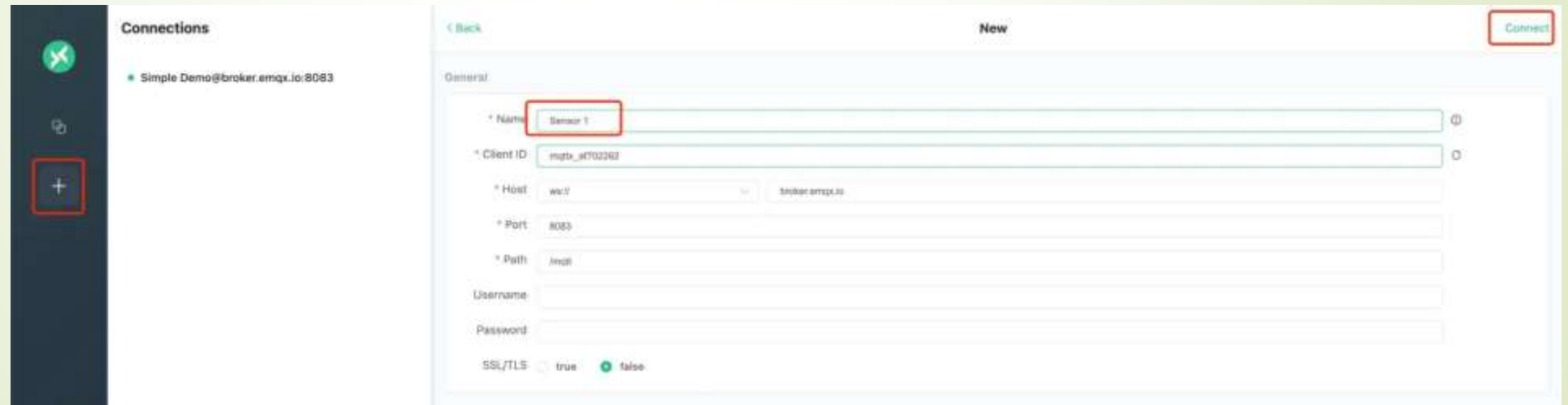
- Nakon uspješnog pretplaćivanja, vidijet ćete dodatni zapis u sredini liste pretplata.



Početak rada sa **MQTT**: Brzi vodič

Objavljivanje MQTT poruka:

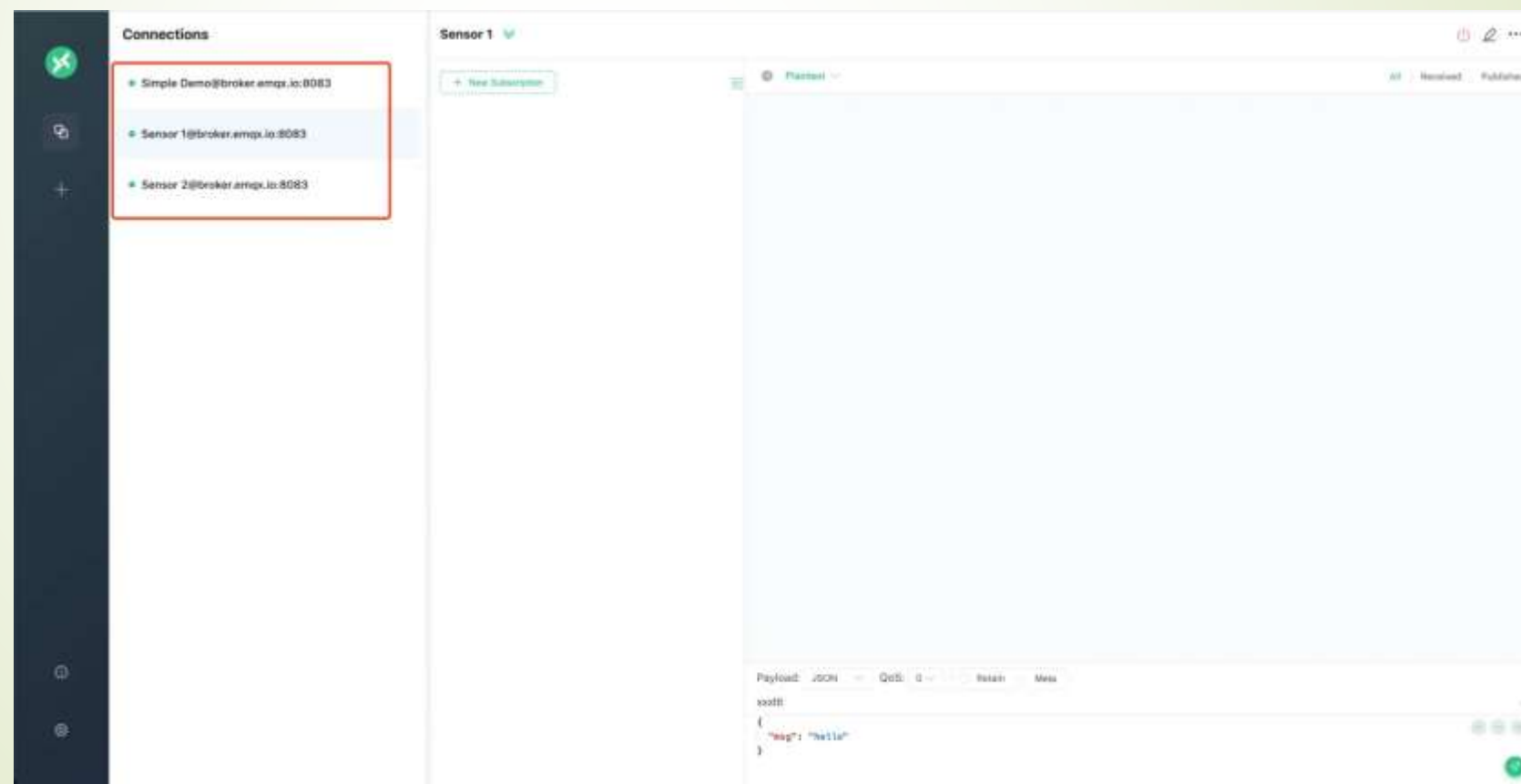
- Zatim treba kliknuti dugme + na lijevom meniju kako bismo stvorili dvije veze, senzor 1 i senzor 2, kako bismo simulirali dva temperaturna senzora.



Početak rada sa **MQTT**: Brzi vodič

Objavljivanje MQTT poruka:

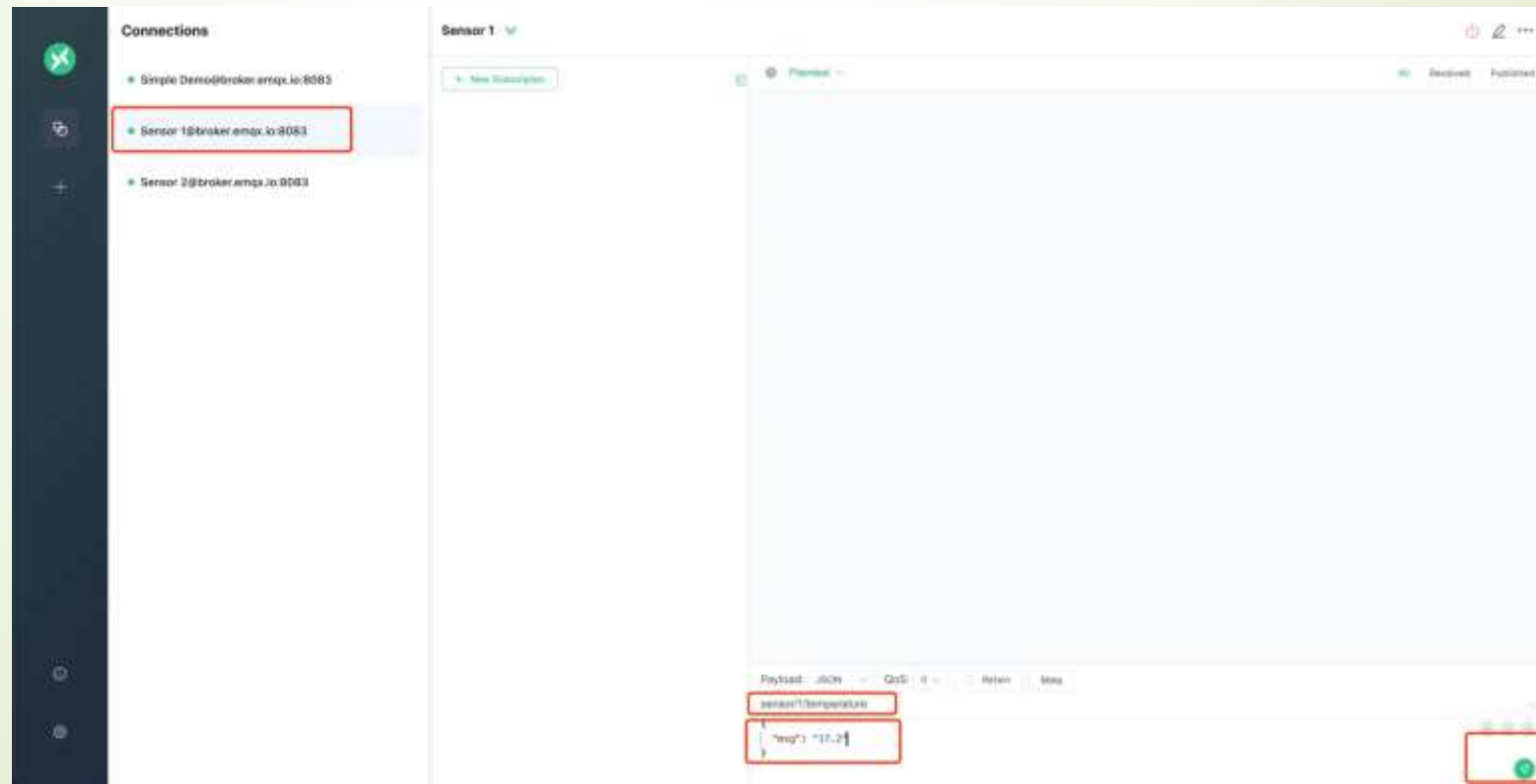
- Nakon što se veza kreira, vidjet ćete tri veze, a sve tačkice mrežnog statusa s lijeve strane veza bit će zelene.



Početak rada sa **MQTT**: Brzi vodič

Objavlјivanje MQTT poruka:

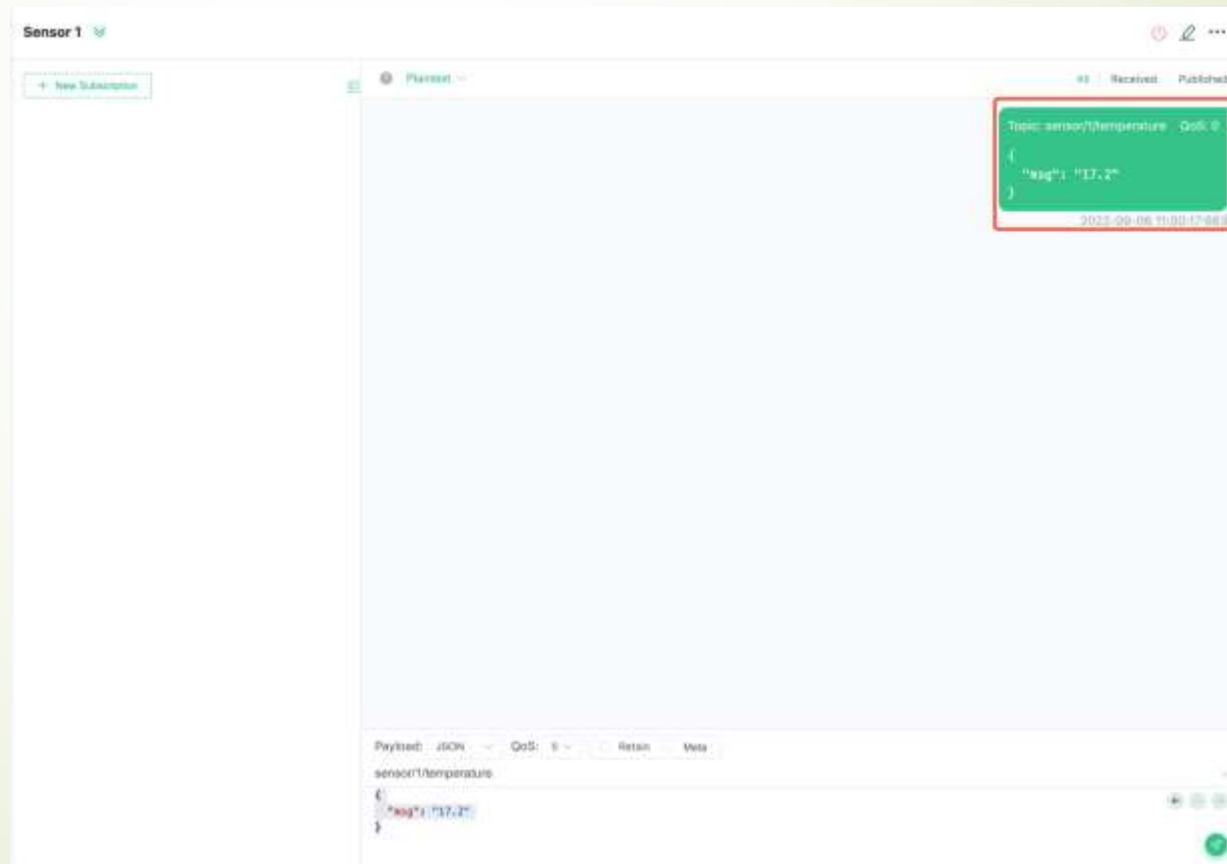
- Izaborite vezu **Senzor 1**, i unesite temu za objavlјivanje **senzor/1/temperatura** u donjem levom delu stranice, unesite sledeću poruku, u formatu JSON, u okvir za poruke i kliknite na dugme za objavlјivanje u donjem desnom uglu da biste poslali poruku.



Početak rada sa **MQTT**: Brzi vodič

Objavljivanje MQTT poruka:

- Poruka je uspješno poslana na sljedeći način.



The screenshot displays the MQTT Explorer interface for a topic named "Sensor 1". The "Published" tab is active, showing a single message. The message content is a JSON object: `{ "temp": "17.2" }`. The message is highlighted with a red box. The interface also shows the topic name, quality of service (QoS), and retention settings. At the bottom, the "Publish" button is visible, and the message content is shown in a text area.

```
Topic: sensor1/temperature  QoS: 0  
{  
  "temp": "17.2"  
}
```

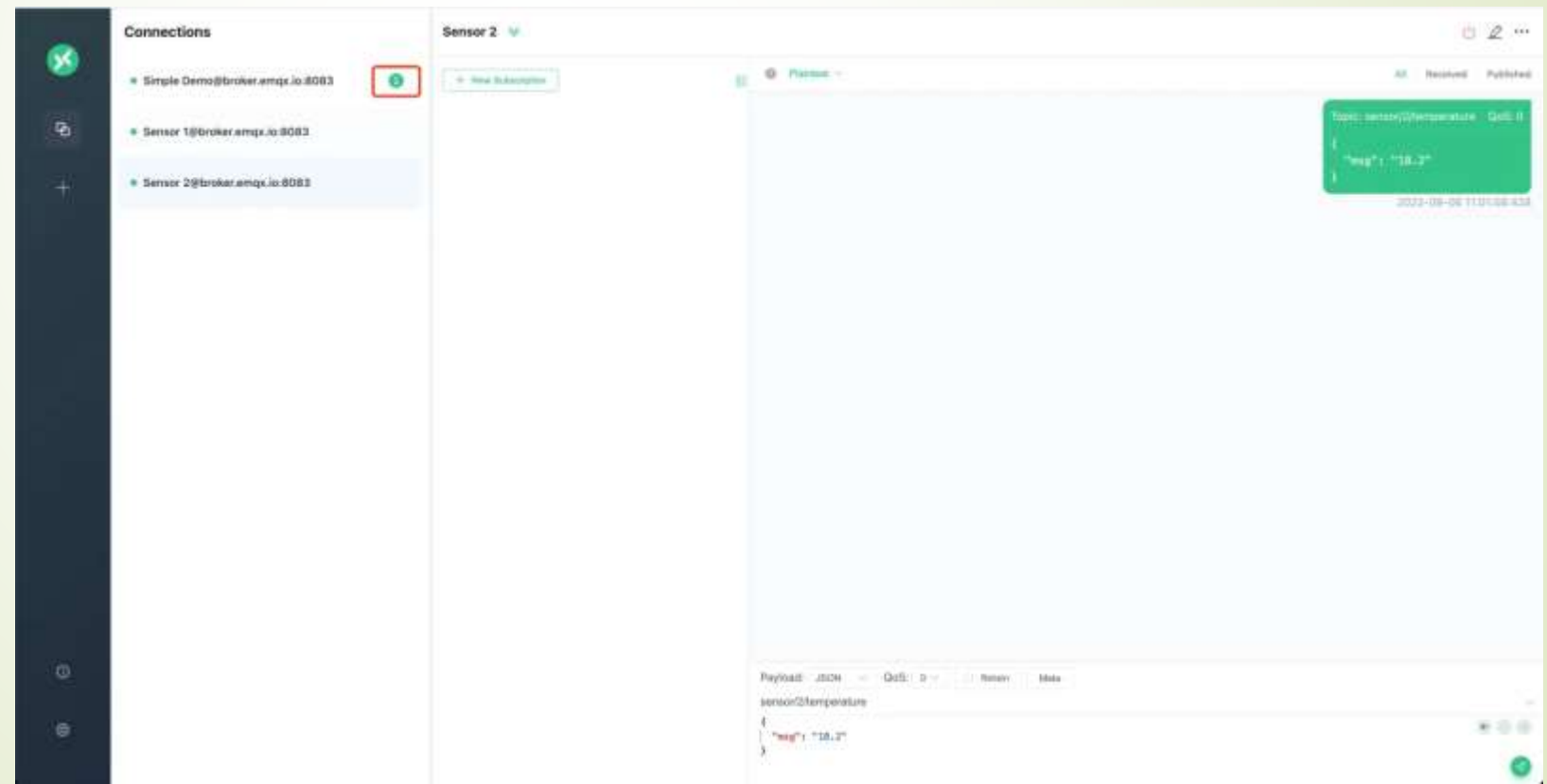
Published: JSON QoS: 0 Retain Web

```
sensor1/temperature  
{  
  "temp": "17.2"  
}
```

Početak rada sa **MQTT**: Brzi vodič

Objavljivanje MQTT poruka:

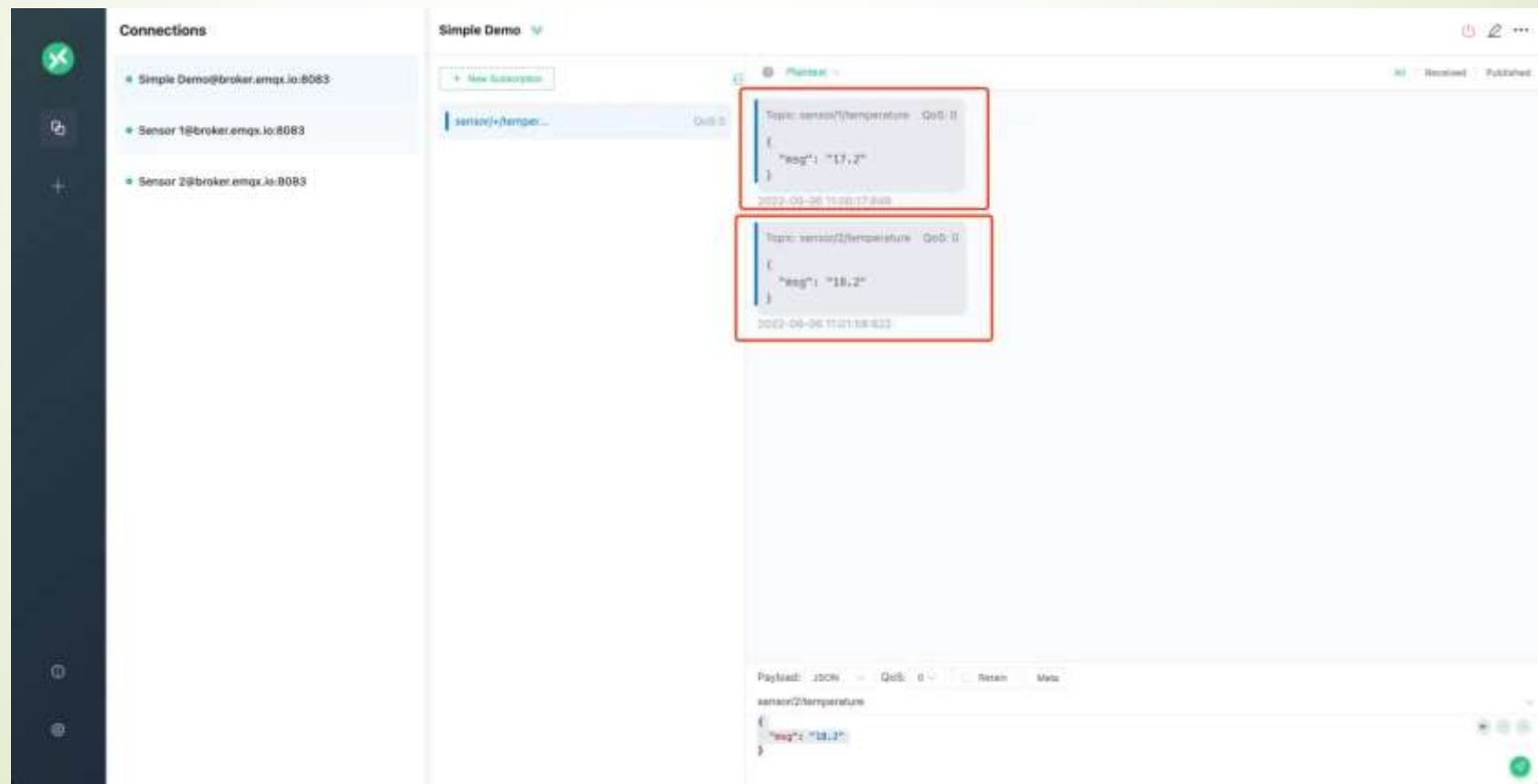
- Koristeći iste korake, objavite sljedeću JSON poruku u temi **senzor/2/temperatura** u vezi senzora 2.



Početak rada sa **MQTT**: Brzi vodič

Objavljivanje MQTT poruka:

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



The screenshot displays the MQTT Explorer interface. On the left, the 'Connections' panel lists three connections: 'Simple Demo@broker.emqx.io:8083', 'Sensor 1@broker.emqx.io:8083', and 'Sensor 2@broker.emqx.io:8083'. The main panel shows a subscription for 'sensor/+/temperature' with QoS 0. Two messages are displayed, both highlighted with red boxes:

- Message 1: Topic: sensor/1/temperature QoS: 0, Payload: {"msg": "17.2"}, Received: 2023-09-06 11:00:17.648
- Message 2: Topic: sensor/2/temperature QoS: 0, Payload: {"msg": "18.2"}, Received: 2023-09-06 11:01:18.623

At the bottom, the 'Payload' field shows the JSON structure: {"msg": "18.2"}.



Arduino uređaj kao klijent

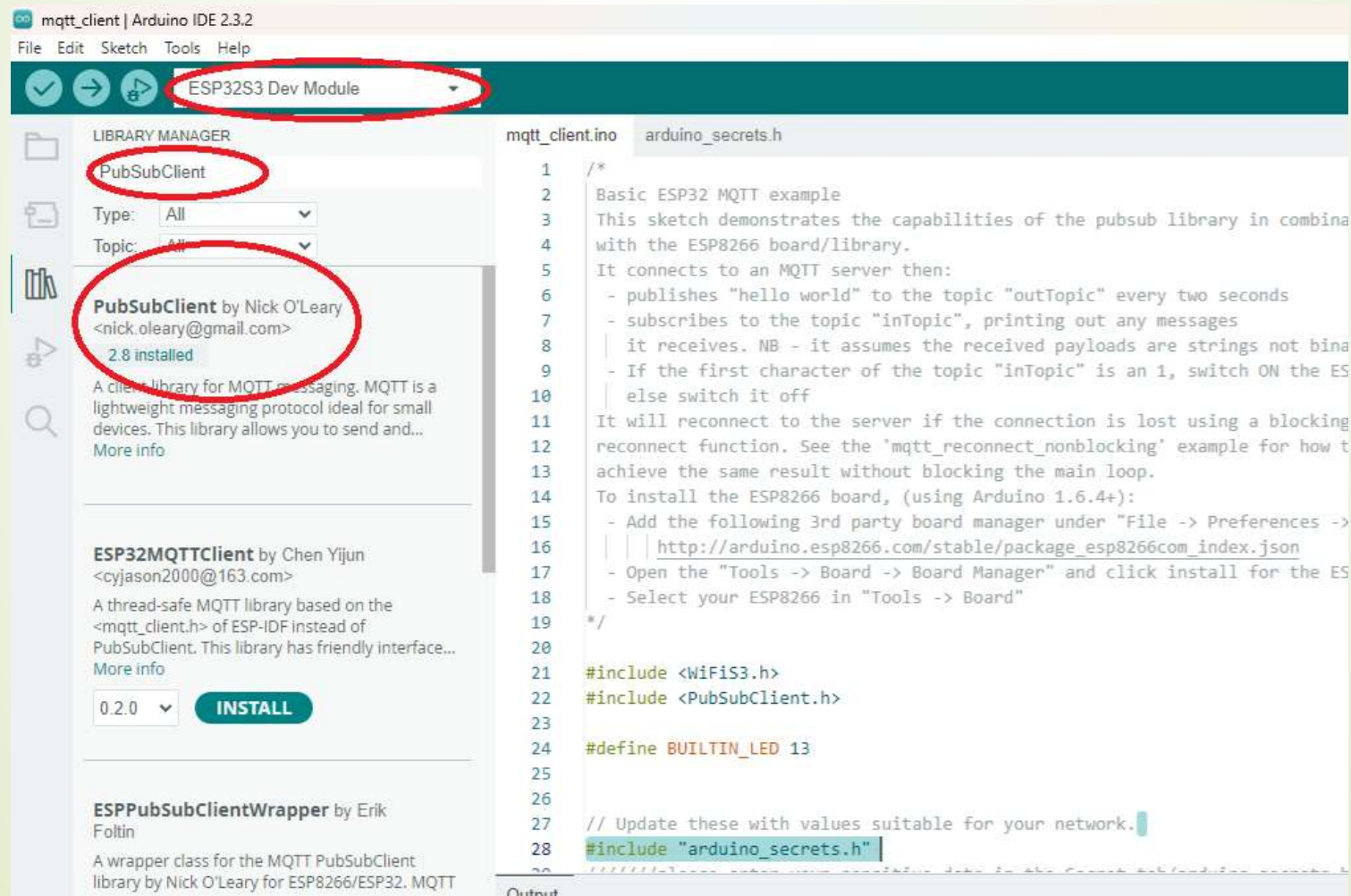
Upišite u Arduino Uno R4 WiFi skeč nazvan:
mqtt_cilent.ino

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:



The screenshot shows the Arduino IDE 2.3.2 interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The board selection dropdown is set to 'ESP32S3 Dev Module'. The Library Manager is open, displaying a search for 'PubSubClient'. The search results show 'PubSubClient by Nick O'Leary' with version 2.8 installed. Below it, 'ESP32MQTTClient by Chen Yijun' is visible with version 0.2.0 and an 'INSTALL' button. The code editor shows the 'mqtt_client.ino' file with the following code:

```
1  /*
2  Basic ESP32 MQTT example
3  This sketch demonstrates the capabilities of the pubsub library in combina
4  with the ESP8266 board/library.
5  It connects to an MQTT server then:
6  - publishes "hello world" to the topic "outTopic" every two seconds
7  - subscribes to the topic "inTopic", printing out any messages
8  | it receives. NB - it assumes the received payloads are strings not bina
9  - If the first character of the topic "inTopic" is an 1, switch ON the ES
10 | else switch it off
11 It will reconnect to the server if the connection is lost using a blocking
12 reconnect function. See the 'mqtt_reconnect_nonblocking' example for how t
13 achieve the same result without blocking the main loop.
14 To install the ESP8266 board, (using Arduino 1.6.4+):
15 - Add the following 3rd party board manager under "File -> Preferences ->
16 | | http://arduino.esp8266.com/stable/package_esp8266com_index.json
17 - Open the "Tools -> Board -> Board Manager" and click install for the ES
18 - Select your ESP8266 in "Tools -> Board"
19 */
20
21 #include <WiFi3.h>
22 #include <PubSubClient.h>
23
24 #define BUILTIN_LED 13
25
26
27 // Update these with values suitable for your network.
28 #include "arduino_secrets.h"
```


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.

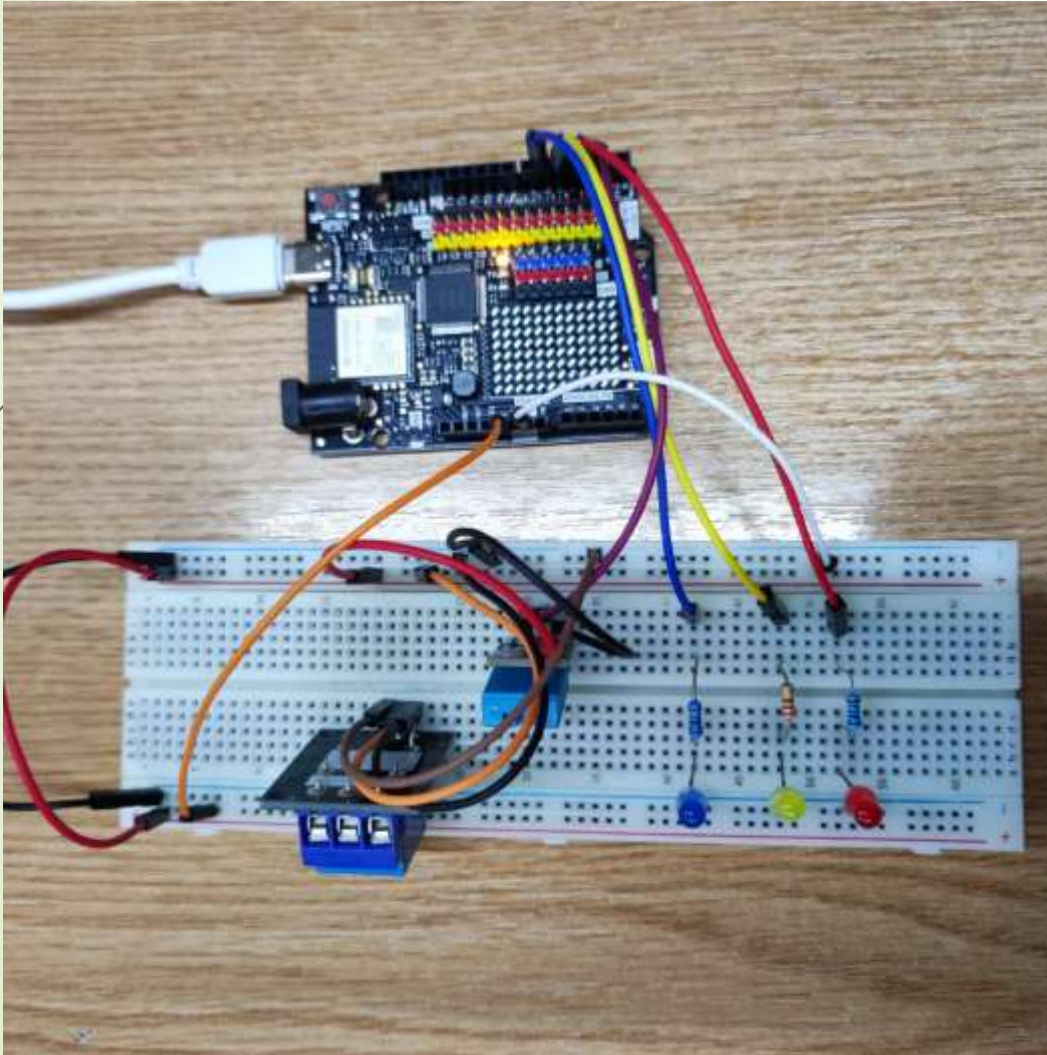
Kada se pretplatite na teme ili objavite poruke koristeći MQTT (Message Queuing Telemetry Transport), mikrokontrolerska ploča mora stalno komunicirati s MQTT brokerom kako bi primala poruke ili slala ažuriranja. Za ovu kontinuiranu komunikaciju odgovorna je funkcija **client.loop()**.

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 220Ω spojena na pin 6.
Žuta LED je preko otpornika 220Ω spojena na pin 5.
Plava LED je preko otpornika 220Ω spojena na pin 3.

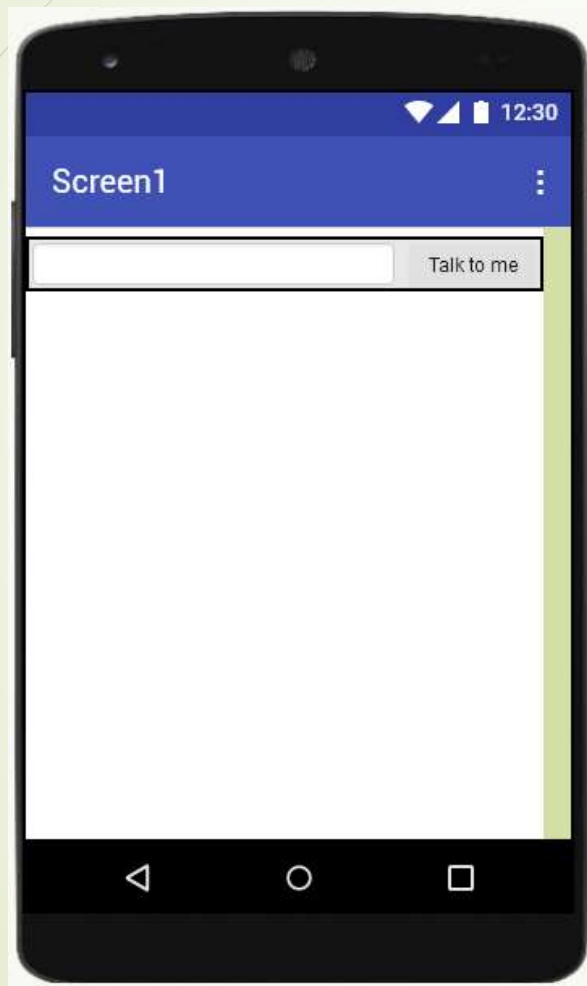
Signal Relay-a je spojen na pin A0. Uz to VCC je spojeno na 5V, a GND na GND.

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>

Prpriprema za vježbu



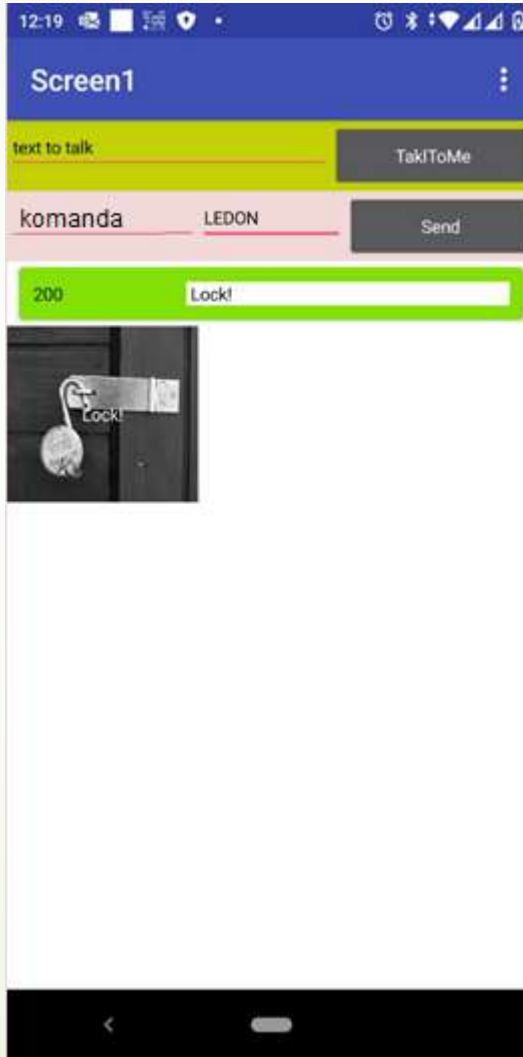
Zajednički ćemo kreirati aplikaciju čiji je izgled prikazan na slici.

Priprema za vježbu



Na osnovnu aplikaciju, iz predhodnog slajda, dodat ćemo i dio koji će omogućiti povezivanje sa našim mikrokontrolerskim dizajnom.

Vježba 1



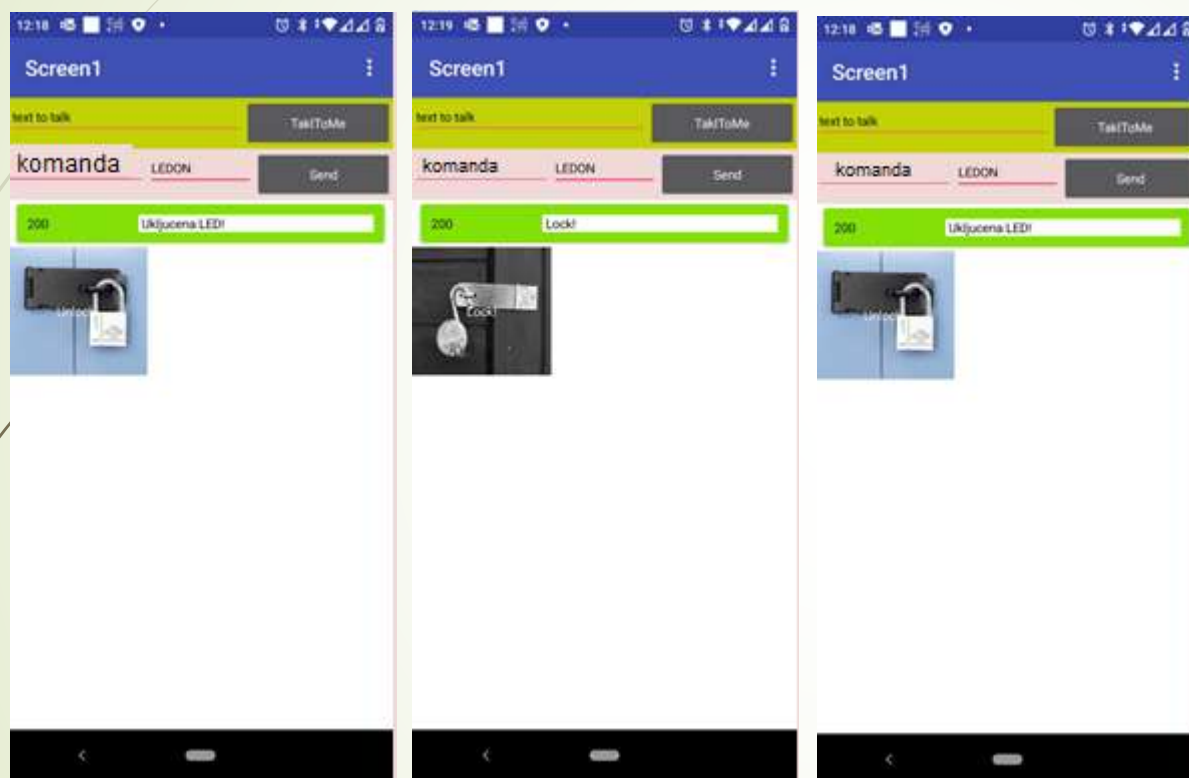
Dopuniti aplikaciju mogućnošću otvaranja brave (odnosno upravljanja rele-om).

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

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

(1 – 0.5 bodova)

Vježba 2

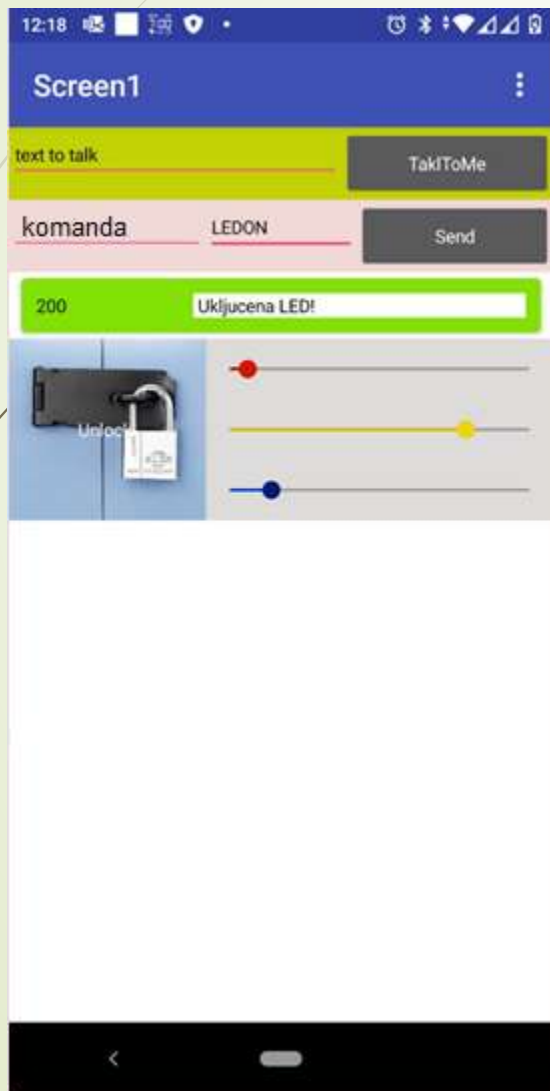


Dopuniti rad sa rele-om, na sljedeći način:

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 istalirana **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 koristi 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



Upravlјati 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



Upravlјati 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“.

(2 - 1 – 0.5 bodova)

U ovom primjeru koristi se I²C LCD. Biblioteka za rad sa njime može se preuzeti sa linka:

https://github.com/johnrickman/LiquidCrystal_I2C