

Node-RED i MQTT protokol

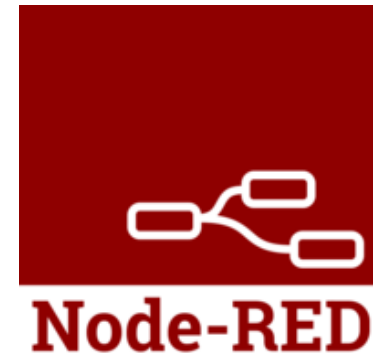


Node-RED

- Node-RED je vizuelni alat za kreiranje tokova podataka
- Razvijen je od strane IBM-a, a od 2016. je open-source projekat
- Omogućava povezivanje hardverskih uređaja, API-ja i online servisa, kao sastavnih delova IoT ekosistema
- Node-RED obezbeđuje kreiranje JavaScript funkcija kroz editor tokova podatka baziran na web pregledaču
- Kreirani tokovi se skladišteni u JSON formatu
- Osnovna terminologija:
 - **Nodovi** (engl. *nodes*) su blokovi koda koji obavljaju određene zadatke
 - Međusobnim povezivanjem nodova dobijaju se **tokovi** (engl. *flows*)

Node-RED između ostalog omogućava na jednostavan način obavljanje sledećih operacija:

- Pristup GPIO pinovima na RPi
- Uspostavljanje MQTT komunikacije sa različitim pločama (Arduino, ESP32 i sl.)
- Kreiranje GUI-ja
- Komunikacija sa spoljnim servisima (IFTTT.com, Adafruit.io, Thing Speak i sl.)
- Preuzimanje podataka sa web servisa pozivom odgovarajućih API-ja
- Tempiranje događaja
- Skladištenje i očitavanje podataka iz baze



Node-RED instalacija

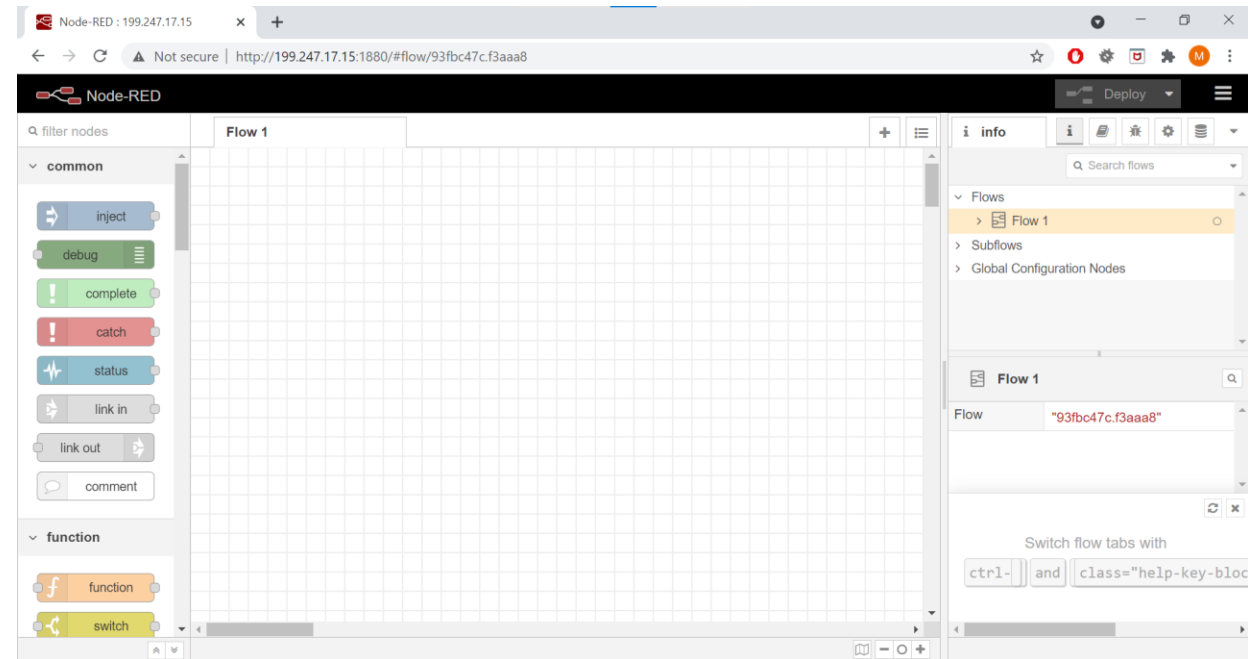
- Primeri koji će biti demonstrirani u ovoj vežbi koriste Node-RED, instaliran na Linux sistemu (RPi u lokalnoj mreži ili Linux server sa statičkom IP adresom)
- Node-RED instalacija:

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```
- Podešavanje automatskog startovanja Node-RED po restartu sistema:

```
$ sudo systemctl enable nodered.service
```
- Po završenoj instalaciji neophodno je restartovati server:

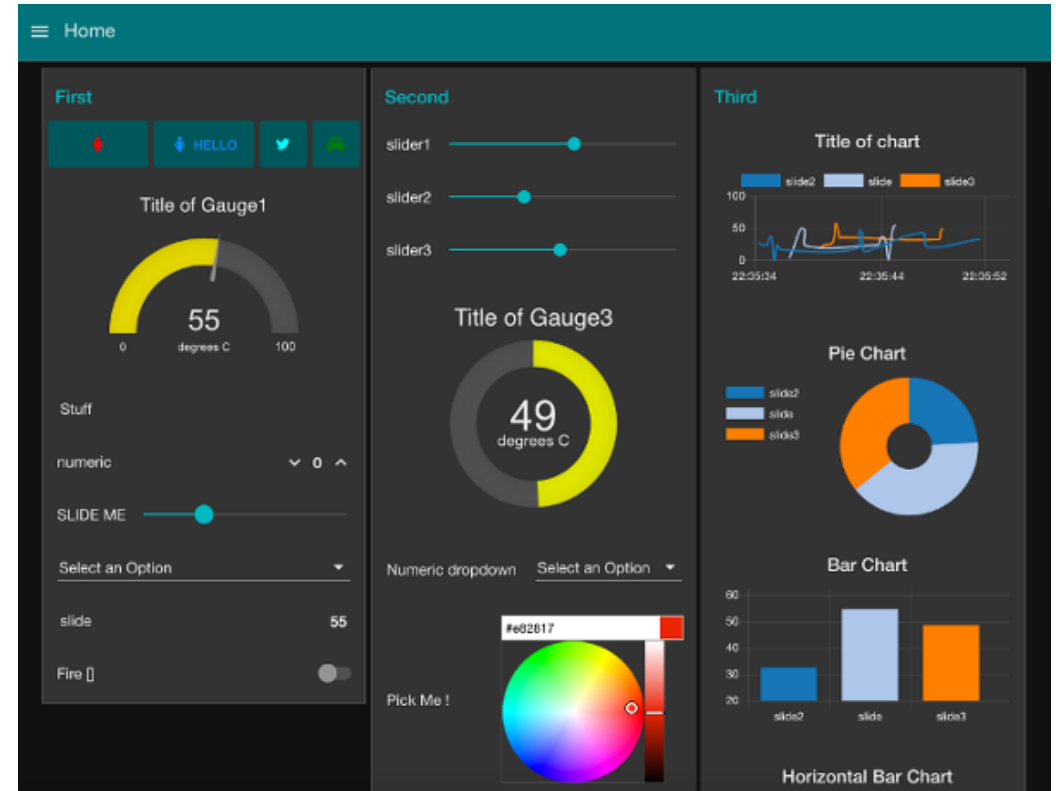
```
$ sudo reboot
```
- Nakon restartovanja, Node-RED web interfejs je dostupan iz web pregledača preko URL:

```
http://IP_ADRESA_SERVERA:1880
```



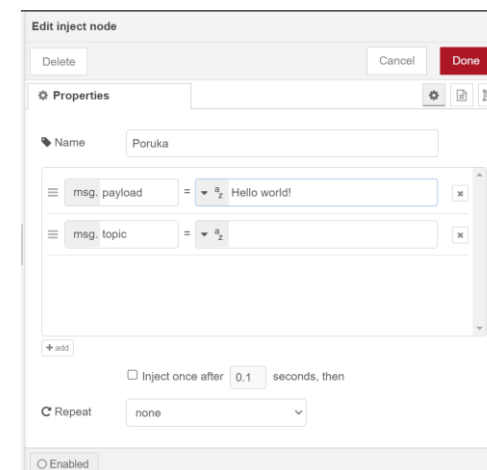
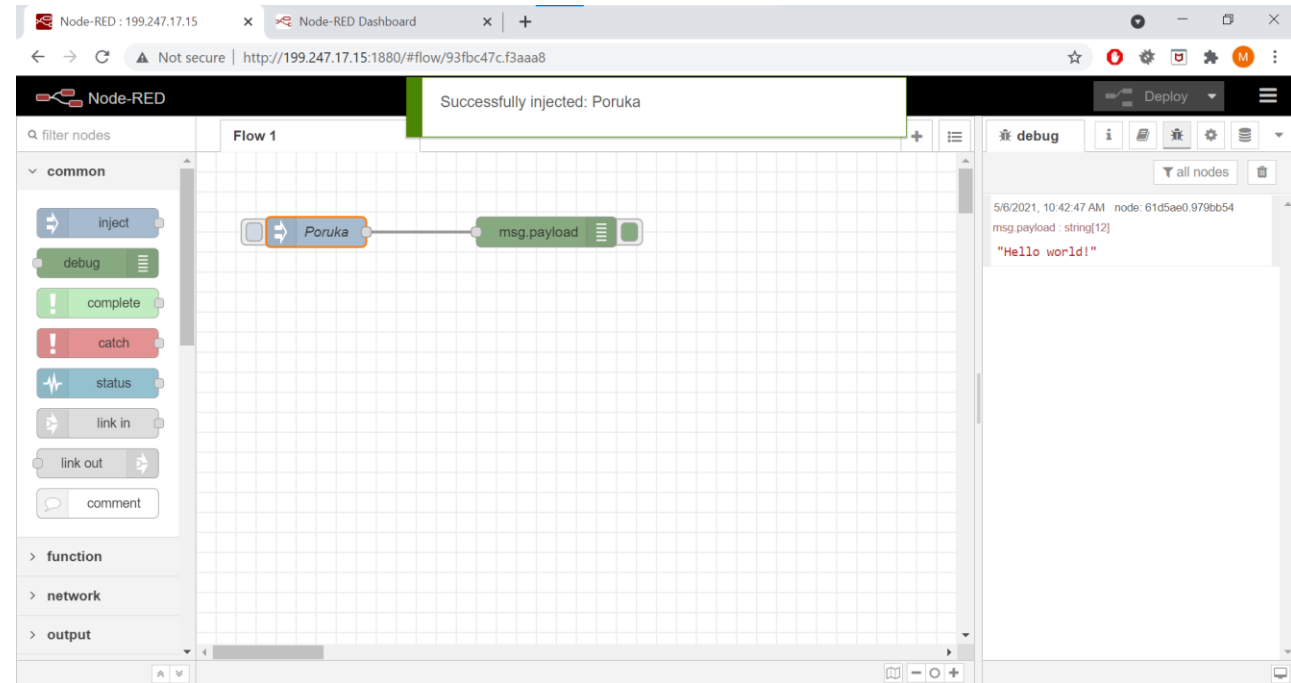
Node-RED dashboard

- Node-RED Dashboard je modul koji obezbeđuje skup nodova za Node-RED koji omogućavaju kreiranje grafičkih korisničkih interfejsa na jednostavan način
- Više informacija dostupno je na zvaničnoj stranici:
<http://flows.nodered.org/node/node-red-dashboard>
- Instalacija:
 - \$ **node-red-stop**
 - \$ **cd ~/.node-red**
 - \$ **npm install node-red-dashboard**
 - \$ **sudo reboot**
- Nakon instalacije, dashboard je dostupan na:
http://IP_ADRESA_SERVERA:1880/ui



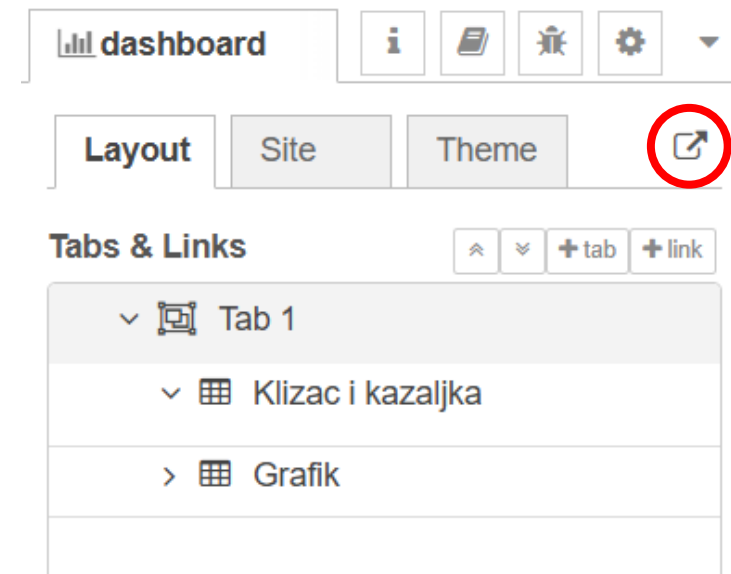
Primer 1: Kreiranje jednostavnog toka podataka

- Za početak će biti testiran jednostavan primer koji ilustruje kreiranje Node-RED toka podataka
- Prvo je potrebno da svaki student kreira sopstveni Node-RED tok (**flow**), pri čemu je naziv toka broj indeksa studenta.
- U meniju **common** potrebno je izabrati **inject** i **debug** nodove i međusobno ih povezati, kao što je prikazano na slici gore desno
- U okviru podešavanja noda inject (otvaranje na dvoklik), postaviti polje **Name** na „Poruka“, tip poruke (**msg.payload**) na string i uneti tekst poruke, kao što je prikazano na slici dole desno
- Nakon toga, pokrenuti kreirani tok pritiskom na taster **Deploy** i sa desne strane izabrati **Debug** tab
- Kada je tok pokrenut, pritiskom tastera sa leve strane inject noda „Poruka“, inicira se slanje poruke koja će biti vidljiva u debug konzoli



Primer 2: Kreiranje korisničkog interfejsa

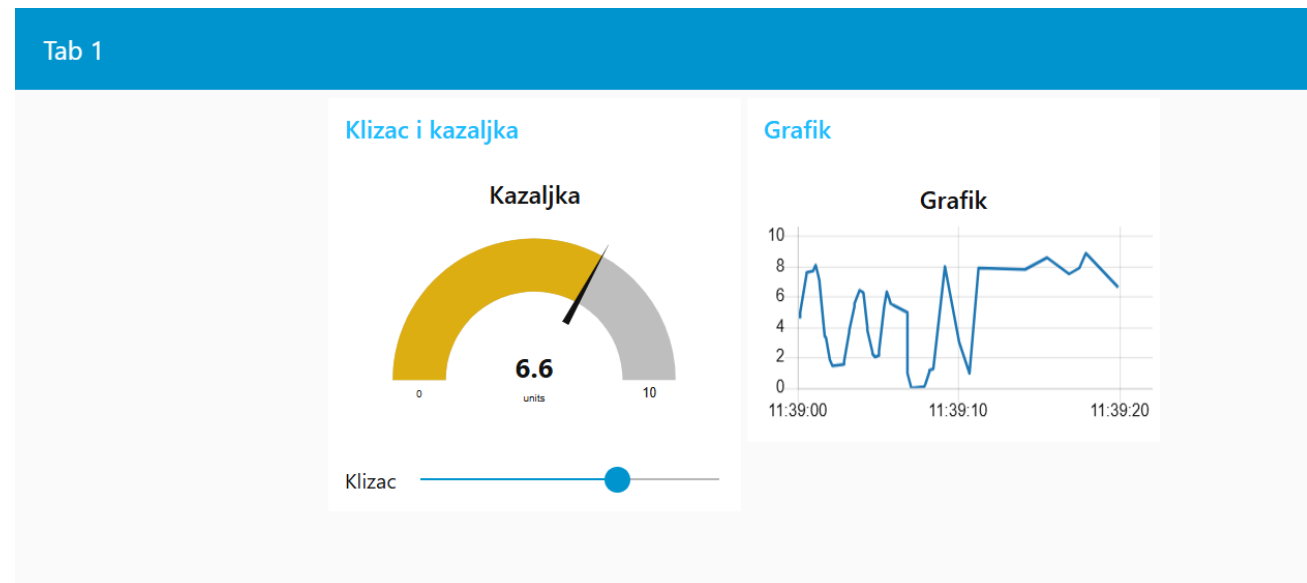
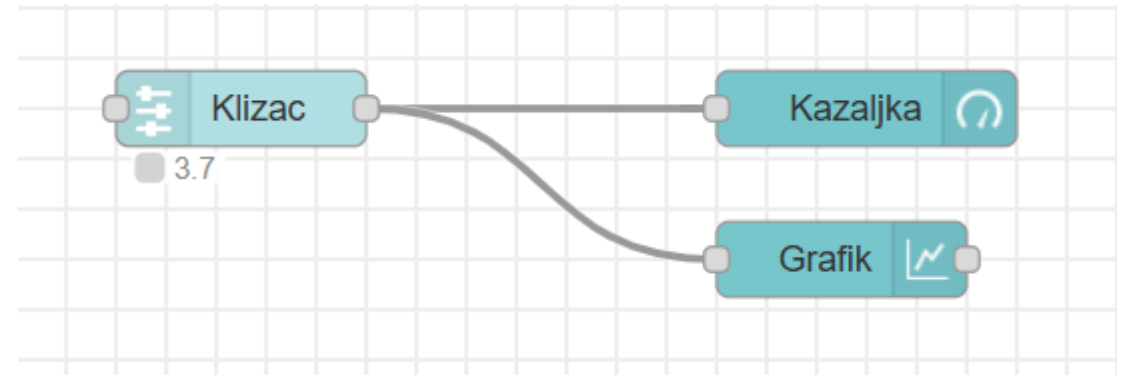
- Za kreiranje grafičkog korisničkog interfejsa koriste se nodovi koji pripadaju meniju **dashboard** i koji predstavljaju grafičke elemente (eng. widgets)
 - Korisnički interfejs se organizuje u tabove i grupe:
 - **Tabovi** su stranice u okviru korisničkog interfejsa, organizovane na sličan način kao tabovi u web pregledaču
 - **Grupe** omogućavaju podelu tabova na sekcije, u cilju lakše organizacije grafičkih elemenata
 - Svaki grafički element mora biti pridružen nekoj od grupa, čime se određuje njegova pozicija u okviru korisničkog interfejsa
- U ovom primeru, potrebno je sa desne strane Node-RED editora izabrati tab **dashboard** i u okviru njega **Layout**
 - Pomoću tastera **+tab** dodati novi tab (Tab 1) i u okviru njega pomoću tastera **+group** dodati nove grupe (Klizac i kazaljka, Grafik)
 - Trenutni izgled korisničkog interfejsa moguće je videti pritiskom na ikonicu označenu na slici



- Na Flow1 radnu površinu dodati 3 grafička elementa iz dashboard menija sa sledećim parametrima i povezati ih kao što je prikazano na slici gore desno:

- Slider: Group -> [Tab1] Klizac i kazaljka; Label -> Klizac; Range: min 0, max 10, step 0.1
- Gauge: Group -> [Tab1] Klizac i kazaljka; Label -> Kazaljka
- Chart: Group -> [Tab1] Grafik; Label -> Grafik; X-axis: last 1 minutes

- Nakon raspoređivanja grafičkih elemenata i pokretanja toka pritiskom na dugme Deploy, potrebno je proveriti njihov izgled i funkcionalnost pokretanjem Dashboard interfejsa (slika dole desno)



MQTT

- MQTT (Message Queuing Telemetry Transport) je protokol aplikativnog sloja zasnovan na publish-subscribe arhitekturi
- Zbog svoje jednostavnosti, često se koristi u IoT aplikacijama gde su prisutni uređaji sa skromnim komunikacionim resursima
- Po svojoj prirodi olakšava komunikaciju između više uređaja u isto vreme

MQTT – How It Works

Send a command to **control an output**



Read and **publish** data



Publish

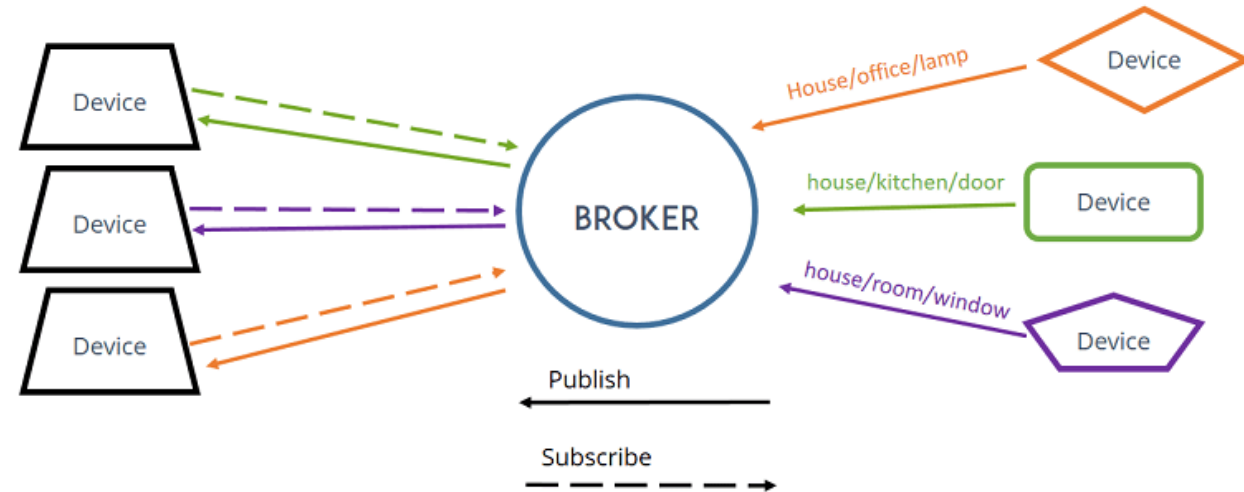


Subscribe



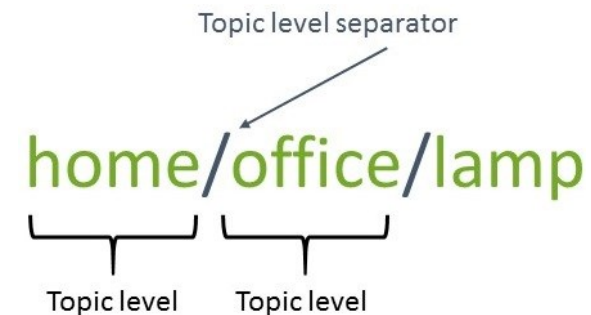
MQTT broker

- **Broker** predstavlja centralni serverski entitet u MQTT sistemu
- U njegove odgovornosti spadaju:
 - Prijem svih poruka
 - Filtriranje poruka
 - Donošenje odluka kome će poruke biti prosleđene, na osnovu toga ko je pretplaćen na temu kojoj pripada poruka
 - Prosleđivanje poruka svim pretplaćenim klijentima
- Komunikacija na relaciji **klijent-broker** (*publisher - broker, odnosno broker - subscriber*) podržava 3 nivoa kvaliteta usluge (Quality of Service):
 - **QoS 0** (najviše jednom) - pošiljalac šalje poruku samo jednom i klijent i broker ne preduzimaju dalje korake u cilju potvrđivanja isporuke (fire and forget)
 - **QoS 1** (najmanje jednom) - pošiljalac će poslati poruku jednom ili više puta, sve dok ne dobije potvrdu o prijemu (acknowledged delivery)
 - **QoS 2** (tačno jednom) - pošiljalac i primalac sprovode handshaking na dva nivoa, kako bi osigurali da je isporučena tačno jedna kopija poruke (assured delivery)



MQTT – osnovni koncepti

- MQTT se zasniva na *publish-subscribe* principu: uređaj može slati (objavljivati) poruku koja pripada određenoj temi, ili može biti pretplaćen na temu i primiti poruke koje joj pripadaju
- Primer:
 - Uređaj 1 objavljuje (publish) poruku na određenu temu
 - Uređaj 2 je pretplaćen (subscribed) na istu temu na koju objavljuje uređaj 1
 - Prema tome, uređaj 2 će primiti sadržaj poruke
- Drugi važan koncept su teme (*topics*)
- Teme su mehanizam za filtriranje i razvrstavanje poruka koje se šalju putem MQTT protokola
- Predstavljaju se stringovima koji su međusobno odeljeni kosim crtama (/), pri čemu svaki deo stringa iza sledeće kose crte predstavlja novi hijerarhijski nivo
- Pri tome treba voditi računa da su teme *case sensitive* (vodi se računa u razlici između velikih i malih slova)



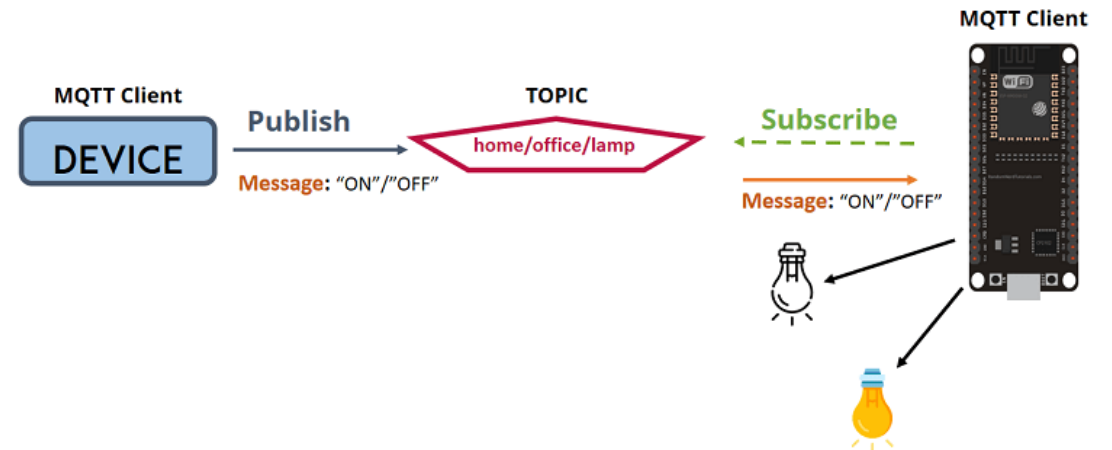
Mosquitto Broker - instalacija

- Instalacija Mosquitto Brokera podrazumeva izvršenje sledećih komandi:
 - \$ sudo apt update**
 - \$ sudo apt install -y mosquitto mosquitto-clients**
 - \$ sudo systemctl enable mosquitto.service**
- Po okončanju instalacije, preporučljivo je izvršiti proveru verzije (trebala bi bito 1.4.x ili viša):
 - \$ sudo systemctl enable mosquitto.service**
- Ukoliko prethodna komanda vrati „Error: Address already in use“, to znači da je broker već aktivan i sve je u redu



MQTT primer: kućni sistem za kontrolu svetla

- Upravljački uređaj (npr. mobilni telefon sa odgovarajućom aplikacijom) objavljuje poruke **“ON”** i **“OFF”** na temu **home/office/lamp**
- Uređaj koji kontroliše svetlo (npr. ESP32) je pretplaćen na temu **home/office/lamp**.
- Kada ESP32 primi poruku **“ON”** ili **“OFF”**, u skladu sa sadržajem poruke uključuje ili isključuje svetlo



Primer 3: MQTT (Node-RED ↔ ESP32)

Kontrola LED (Node-RED → ESP32)

- Napraviti u Node-RED dashboardu grupu **LED kontrola** sa sledećim grafičkim elementima:
 - **Button „ON“**, za uključivanje LED
 - **Button „OFF“**, za isključivanje LED
 - **Text**, za prikaz trenutnog stanja LED diode
 - **Mqtt out**, publish na temu “ESP32/led”

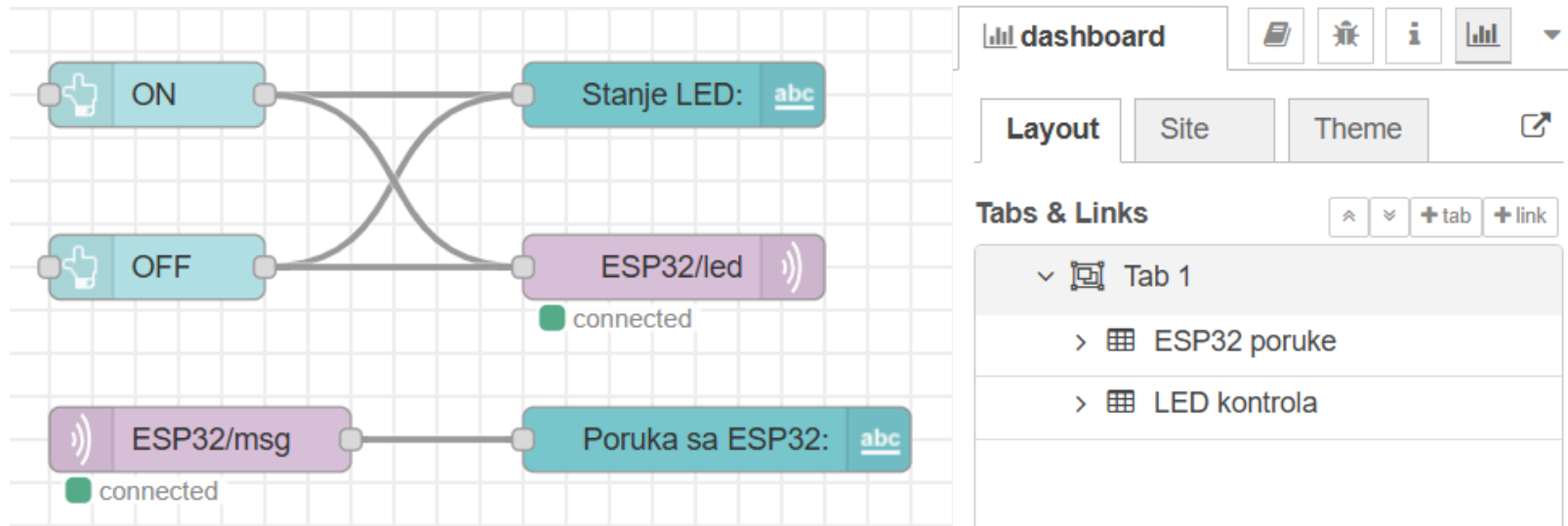
Prikaz poruka (ESP32 → Node-RED)

- Napraviti u Node-RED dashboardu grupu **ESP32 poruke** sa sledećim grafičkim elementima:
 - **Mqtt in**, subscribe na temu “ESP32/msg”
 - **Text**, za prikaz pristiglih poruka

ESP32 aplikacija

- Subscribe na “ESP32/led” – upravljanje LED (ON/OFF)
- Publish na “ESP32/msg” – slanje poruka ka Node-RED
- Modifikovati primer: Examples -> PubSubClient -> **mqtt_esp8266**
- *mqtt_esp32.ino*
- Dokumentacija: <http://pubsubclient.knolleary.net/api>

Primer 3: MQTT (Node-RED ↔ ESP32)



ESP32 poruke	LED kontrola
Poruka sa ESP32: hello world #57	<input type="button" value="OFF"/>
	Stanje LED: ON
	<input type="button" value="ON"/>

Zadatak

- Realizovati IoT sistem u kome Node-RED i ESP32 komuniciraju putem MQTT protokola u oba smera.
- Na Node-RED dashboardu omogućiti unos granične vrednosti temperature (threshold), koja se šalje ESP32 uređaju.
- ESP32 treba da očitava temperaturu i vlažnost sa senzora, i da na osnovu primljenog praga proverava da li temperatura prelazi zadatu vrednost. U skladu sa tim, određuje status sistema (npr. „OK“ ili „ALARM“).
- Izmerene vrednosti temperature i vlažnosti, kao i status sistema, ESP32 šalje nazad ka Node-RED-u, gde se sve informacije prikazuju na dashboardu u realnom vremenu.