

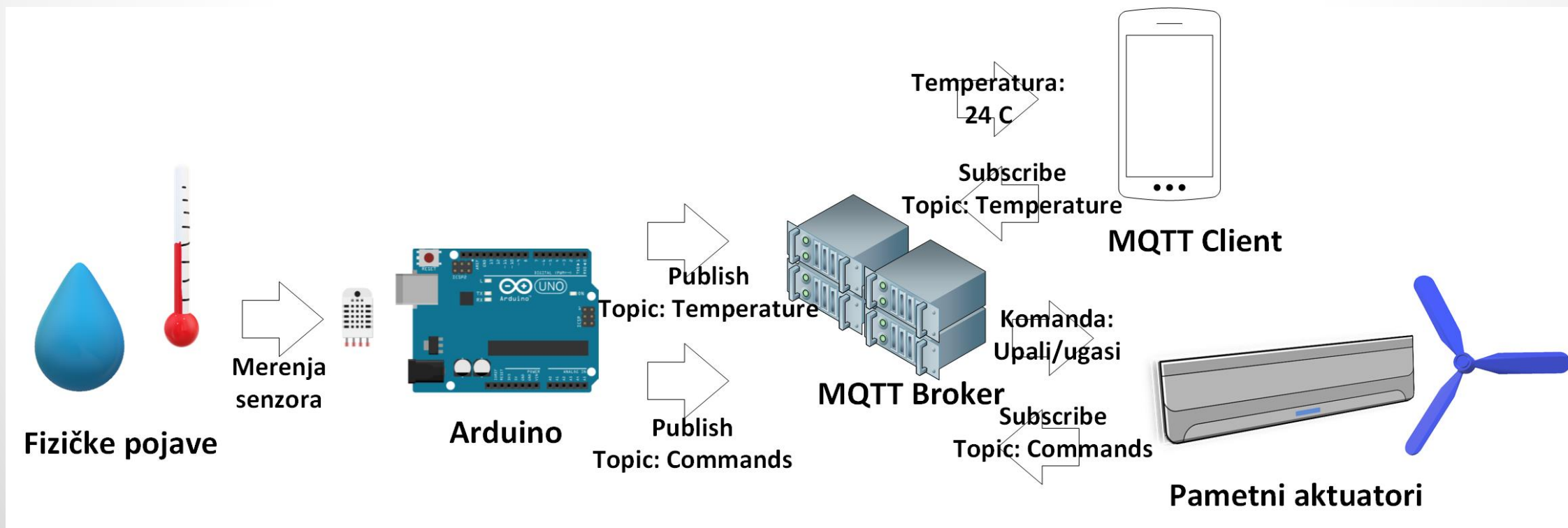
Arduino R4 i MQTT preko WiFi

Internet stvari 2024.

Nenad Petrović

Univerzitet u Nišu, Elektronski fakultet

nenad.petrovic@elfak.ni.ac.rs, kancelarija 323

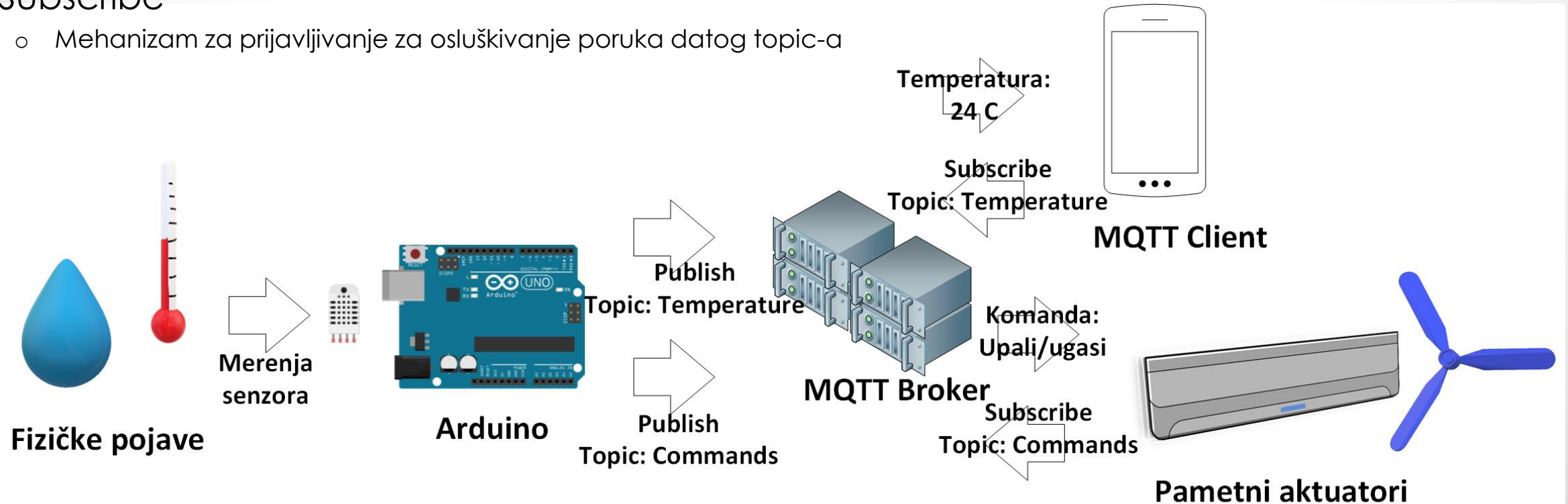


Uvod

- Message Queuing Telemetry Transport (MQTT) je publish-subscribe protokol za prenos podataka namenjen manjim IoT uređajima
 - Male dimenzije
 - Mala potrošnja
 - Ograničena procesorska moć
- Mali zahtev resursa i nizak overhead komunikacije
 - Zaglavlja malih dimenzija sa ciljem povećanja protoka
- Zahteva mrežnu konekciju i povezanost na internet
- Arduino nema ugrađen mrežni čip, pa zahteva dodatke
 - Ethernet Shield
 - **WiFi čip – ESP32-S3**
- U nastavku ćemo videti primer kako Arduino Uno R4 može vršiti razmenu poruka u oba smera preko MQTT protokola korišćenjem ugrađenog ESP32-S3 i MQTT biblioteke – ArduinoMqttClient

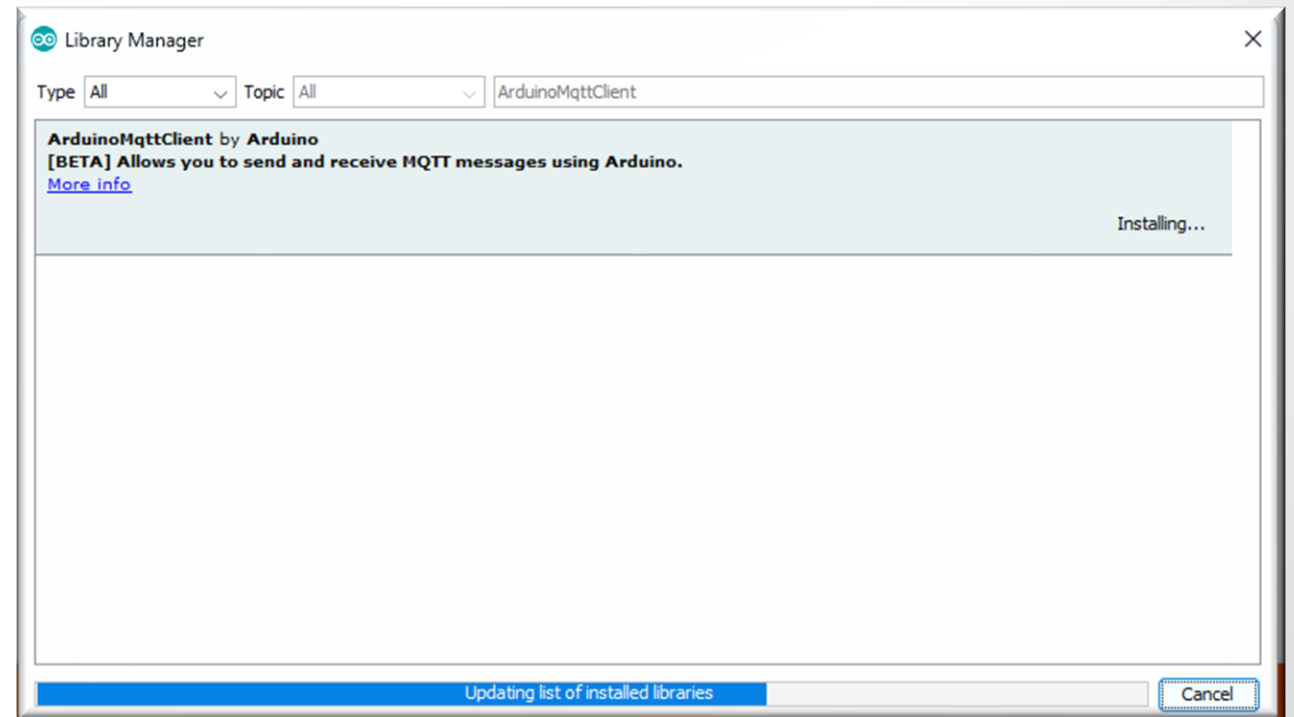
Pregled MQTT arhitekture – Publish/Subscribe

- Broker
 - Server koji služi kao posrednik u razmeni poruka
 - Ne čuva trajno podatke, samo ih prosleđuje pretplaćenim klijentima
- Topic
 - Predstavlja string na osnovu koga razlikujemo značenje i ulogu poruke
- Publish
 - Slanje poruke svima koji osluškiju dati topic
- Subscribe
 - Mehanizam za prijavljivanje za osluškivanje poruka datog topic-a



Instalacija ArduinoMqttClient u Arduino IDE

- Preduslov: instalirati podršku za ploču
 - ***Tools->Board->Boards Manager->"Arduino R4 Boards"***
- Potrebno je MQTT biblioteku
 - ***Sketch->Include Library->Manage Libraries...***
 - Ukucati naziv biblioteke – ***ArduinoMqttClient***
 - Klik na Install
- Na početku programa
 - ***#include <ArduinoMqttClient.h>***
- MQTT + WiFi (ESP32-S3)
 - ***#include <WiFiS3.h>***



Inicijalizacija neophodnih promenljivih

- Podesiti naziv i šifru mreže
 - `char ssid[] = "moja_mreze";`
 - `char pass[] = "moja_sifra";`
- Podesiti adresu brokera, bez http prefix-a i slash simbola
 - `const char* server = "test.mosquitto.org";`
- Kreirati WiFi i MQTT objekte
 - `WiFiClient wifiClient;`
 - `MqttClient mqttClient(wifiClient);`
- Deklaracija prototipa metoda za callback
 - `void prijemPoruke(int messageSize);`

Tok programa

- Započeti WiFi konekciju - SETUP
 - **WiFi.begin(ssid, password):** ssid – naziv mreže, password – šifra
- Poželjno sačekati par sekundi da se poveže – obično oko 10s - SETUP
 - **delay(10000);**
- Opciono – username i password - SETUP
 - Username i password: **mqttClient.setUsernamePassword("username", "password");**
 - Jedinstveni ID klijenta: **mqttClient.setId("clientId");**
- Povezati se na MQTT broker za razmenu poruka - SETUP
 - **mqttClient.connect(broker, port);**
 - server – adresa brokera
 - port – default za MQTT je 1883
- Prijaviti se na teme poruka koje se osluškuju - SETUP
 - **mqttClient.subscribe(topic);**
- Slanje poruka – NAKON CONNECT
 - **mqttClient.beginMessage(topic);** - započni poruku
 - **mqttClient.print(vrednost)** – upis sadržaja;
 - **mqttClient.endMessage();** - zatvori poruku
- Osluškivanje poruka prozivkom – LOOP
 - **mqttClient.poll();**

Rukovanje primljenim porukama


- Defnišemo callback funkciju pod nazivom `prijemPoruke`
- U setup-u neophodno dodati
 - `mqttClient.onMessage(onMqttMessage);`
- Poziva se svaki put kada dođe poruka sa MQTT brokera

```
void prijemPoruke(int messageSize) {  
    // Čitamo sadržaj kao string  
    String message = mqttClient.readString();  
  
    //Čitamo topic  
    String topic = mqttClient.messageTopic();  
  
    // Provera tipa poruke  
    if(topic == "ciljani_string1") {  
        obradaPoruke1(message);  
    } else if (message == "ciljani_string2") {  
        obradaPoruke2(message);  
    } else {  
        // Ako se ne poklapa sa predefinisanim topics, print u debug  
        Serial.print("Message to My ESP32");  
        Serial.println(message);  
    }  
    Serial.println();  
}
```

Javni test broker

- HiveMQ
 - <https://www.hivemq.com/public-mqtt-broker/>
- Demo
 - <https://www.hivemq.com/demos/websocket-client/>
- Skup funkcionalnosti
 - Publikovanje poruka - Publish
 - Prijem poruka - Messages
- Dostupan javno online
 - Broker: broker.hivemq.com
 - TCP Port: 1883
 - Websocket Port: 8000

Osluškivanje poruka

Color 

QoS

Topic

Subscribe

Konektovanje klijenta

Connection

Host Port ClientID

Username Password Keep Alive SSL ☐

Last-Will Topic Last-Will QoS Last-Will Retain ☐

Last-Will Message

Connect

Publish **Subscriptions**

Messages

Slanje poruke

Publish

Topic QoS Retain ☐

Message

Publish

Live demo

- Dvosmerna MQTT komunikacija
 - Arduino sa virtuelnim senzorom šalje vrednosti na topic_send
 - Arduino prime poruke od strane MQTT brokera na topic_receive