



**УНИВЕРЗИТЕТ У НОВОМ САДУ**  
**ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**



Петар Зечевић

**Једна реализација комуникације  
централног контролера са крајњим  
уређајима и корисником у интернет  
ствари систему**

Нови Сад, 2019.

# Sadržaj

1. O sistemu
2. Protokoli
3. Krajnji uređaj
  - 3.1. Opis
  - 3.2. Stanje
4. Centralni kontroler
  - 4.1. Sistem topika
5. Korisnik
  - 5.1. Komande
6. Pretpostavke
7. Reference

## **1. O sistemu**

Komponente sistema su:  
krajnji uređaj, centralni kontroler i korisnik.

Centralni kontroler i krajnji uređaji ne znaju ništa unapred jedni o drugima, već samo poznaju format poruka koje mogu da razmenjuju.

Poruke u komunikaciji su predstavljene korišćenjem JSON(Java Script Object Notation) struktura.  
Glavna ideja realizacije je da se na dinamičan način krajnji uređaji, koji predstavljaju senzore i aktuatore povezuju na centralni kontroler i da korisnik može da im pristupa preko centralnog kontrolera.

Pristupati znači da može da proveriti njihovo stanje, tj. vrednosti svih parametara, informacije o uređaju i menja vrednosti parametara.

## 2. Protokoli

### *SSDP - Simple Service Discovery Protocol*

Protokol se koristi radi dinamičkog otkrivanja uređaja u lokalnoj mreži, u našem slučaju krajnji uređaj otkriva koji se centralni kontroler nalazi u okolini.

Krajnji uređaj koji želi da se prikači na kontroler pretražuje mrežu sa *MSEARCH* porukama, a centralni kontroler na *MSEARCH* poruke odgovara sa *RESPONSE* porukama u kojima se na određeno polje postavlja ip adresa i broj porta mqtt brokera kao i naziv topika na koji uređaji mogu da se loguju, tj. budu registrovani od strane centralnog kontrolera.

#### *MSEARCH poruka:*

```
M-SEARCH * HTTP/1.1
HOST:239.255.255.250:1900
MAN:"ssdp:discover"
MX:1
ST:IOTGATEWAY
USER-AGENT:OS/version product/version
```

*RESPONSE* poruka:

HTTP/1.1 200 OK  
CACHE-CONTROL:max-age=120  
DATE:  
EXT:  
LOCATION: brokerIp : brokerPort  
SERVER:OS/version product/version  
ST:IP  
USN: gatewayId  
SM\_ID: userId/gateway/gatewayId/log  
DEV\_TYPE:IOTGATEWAY

Više o protokolu [1].

## *MQTT – Message Queuing Telemetry Transport*

Protokol aplikativnog sloja, najzastupljeniji u IOT sistemima. Koristi se za razmenu informaciju među komponentama sistema nakon sto se prvo ušpešno povežu.

Bazira se na konceptu pretplate i objave koji se vezuju za topik. Kada uređaj objavi poruku na topik, ta poruka stiže svim uređajima koji su pretplaćeni na topik. Komponenta zadužena za prijem i slanje sa topika se naziva broker.

Sistem topika preko kojih komuniciraju komponente je opisan u odeljku centralnog kontrolera.

Više o protokolu [1].

### **3. Krajnji uređaj**

Krajnji uređaji se povezuju na centralni kontroler tako što pretražuju mrežu slanjem multikast paketa uspomoc SSDP protokola.

Ako je centralni kontroler povezan na istu mrežu kao i uređaj, onda mu šalje odgovor korišćenjem istog protokola kako bi mogao da otpočne proces registracije uređaja i njihove međusobne komunikacije.

Ako krajnji uređaj ne pronade centralni kontroler u okviru svoje mreže nakon određenog broja pokušaja, onda prelazi u neaktivno stanje koje traje unapred definisani vremenski period i nakon toga opet pokušava da se poveže sa centralnim kontrolerom.

Nakon sto se uredjaji uspesno registruju, centralni kontroler komunicira sa njima uspomoc MQTT.

Uređaji periodično šalju centralnom kontroleru izmerene vrednosti svojih parametara, s tim što se šalju samo one vrednosti koje su različite u odnosu na prethodno izmerene, radi smanjenja opterećenja centralnog kontrolera.

Osobine krajnjeg uređaja opisuju dve JSON strukture, a to su opis uređaja i stanje uređaja. Opis se sastoji od: grupe uređaja, identifikacije uređaja, servisa i karakteristika svakog parametra koji definiše krajnji uređaj.

Stanje se sadrži od parova parametar-vrednost.

### 3.1 Opis

Primer opisa uređaja:

```
{
  "id" : "light_bulb_1",
  "group" : "bulb",
  "ContactService" :
  {
    "State" : "T/F"
  },
  "LightService" :
  {
    "DimLevel" : "0~100",
    "State": "ON/OFF",
    "Color": "RGB/HS"
  }
}
```

Za parametre imamo dve vrste liste vrednosti a to su opseg i niz.

Opseg: startValue ~ endValue.

Niz: value1/value2/.../valueN

Parametri koji mogu samo da se pročitaju, ali ne i da se modifikuju od strane korisnika su obeleženi tako što se pored njihove liste vrednosti dodaje “|r” string.

Opis jednog multifunkcionalnog senzora

```
{
  "id" : "weather_sensor_1",
  "group" : "sensors",
  "TempService" :
  {
    "Temperature" : "-50~50|r",
  },
  "HumidityService":
  {
    "Humidity" : "900~1100|r"
  }
}
```

## 3.2 Stanje

Primer stanja uređaja:

```
{
  "id" : "light_bulb_1",
  "group" : "bulb",
  "ContactService" :
  {
    "State" : "T"
  }
  "LightService" :
  {
    "DimLevel" : "55",
    "State": "ON",
    "Color": "RGB"
  }
}
```



## **4. Centralni kontroler**

Kada se krajni uređaj odjavi(ugasi ili napusti mrežu), centralni kontroler dobija obaveštenje o tome i uklanja ga iz liste registrovanih uređaja, na isti način i krajnji uređaj zna kada se centralni kontroler odjavio, kako bi mogao da ponovo započne proces pretrage. To je obezbeđeno MQTT protokolom.

Predstavlja most u komunikaciji između korisnika i krajnjih uređaja.  
Skladišti informacije o opisu i stanju svakog registrovanog uređaja.

Korisnik ne komunicira direktno sa uređajima već sve informacije o uređajima dobija od centralnog kontrolera, koji to čuva u svojoj internoj bazi.

Na istom uređaju na kom se pokreće softver za centralni kontroler, pokreće se i mosquitto mqtt broker, više o tome u dokumentu pokretanje.

### **4.1 Sistem topika**

Šablon za topike:

*user\_id/device\_group/device\_id/method.*

Sledi opis topika koje koristi svaka komponenta u formi:  
naziv\_topika ~ pretplata/objava ~ namena topika

### **Topici koje koriste krajnji uređaji:**

*user\_id/device\_group/device\_id/update* ~ pretplata ~  
promena stanja uređaja

*user\_id/device\_group/device\_id/report* ~ objava ~ slanje  
informacija o stanju uređaja

*user\_id/device\_group/device\_id/get* ~ pretplata ~ zahtev  
za stanje uređaja

*user\_id/device\_group/device\_id/status* ~ pretplata ~  
uspešnost logovanja

### **Topici koje koristi centralni kontroler:**

*user\_id/gateway/gateway\_id/log* ~ pretplata ~ registracija  
krajnjih uređaja

*user\_id/gateway/gateway\_id/command* ~ pretplata ~  
obrada korisničkih komandi

*user\_id/gateway/gateway\_id/command/response* ~ objava  
~ rezultati korisničkih komandi

*user\_id/device\_group/device\_id/update* ~ objava ~ slanje  
novih vrednosti parametara na krajnji uređaj

*user\_id/device\_group/device\_id/report* ~ pretplata ~  
prijem i skladištenje stanja krajnjeg uređaja

*user\_id/device\_group/device\_id/get* ~ objava ~ zahtev za  
stanjem krajnjeg uređaja

### **Topici koje koristi korisnik:**

*user\_id/gateway/gateway\_id/command* ~ objava ~ slanje  
komande

*user\_id/gateway/gateway\_id/command/response/info* ~  
pretplata ~ prijem opisa uređaja

*user\_id/gateway/gateway\_id/command/response/state* ~  
pretplata ~ prijem stanja uređaja

## **5. Korisnik**

Korisnik sistema pristupa krajnjim uređajima, tako što šalje komande centralnom kontroleru. Pruža CLI(command line interface) korisniku za rukovanje.

Kroz interfejs korisnik može da:

- Pregleda opis svih registrovanih uređaja
- Pregleda stanje svih registrovanih uređaja
- Zadaje komande centralnom kontroleru

### **5.1 Komande**

Komande su definisane kao namenski jezik, za koji je napravljen interpreter na korisničkoj strani.

Nakon što se utvrdi ispravnost komandi, one se šalju centralnom kontroleru uspomoc MQTT protokola.

Rezultat izvršenja komandi centralni kontroler šalje korisniku korišćenjem istog protokola.

Sledi opis podržanih komandi u formatu:  
sintaksa komande ~ svrha komande.

*SET device\_id.service.param value* ~ postavlja parametar uređaja na zadatu vrednost.

*GET device\_id.info* – zahtev za opis datog uređaja.

*GET device\_id.state* – zahtev za stanje datog uređaja.

*GET \*.info* – zahtev za opis svih registrovanih uređaja.

*GET \*.state* – zahtev za stanje svih registrovanih uređaja.

Stanja i opis krajnjih uređaja koje korisnik dobavi od centralnog kontrolera se skladište na strani korisnika u njegovoj bazi podataka.

Nakon što se komanda uspešno obradi ona se šalje centralnom kontroleru kao JSON struktura u sledećem formatu:

```
SET komanda ->
{
  "command_type" : "SET",
  "group" : "device_group",
  "device" : "device_id",
```

```
    "service" : "service_name",  
    "parameter" : "parameter_name",  
    "value" : "parameter_value"  
}
```

GET komanda →

```
{  
    "command_type" : "GET",  
    "json" : "state | info",  
    "device" : "* | device_id"  
}
```

## **6. Pretpostavke**

Pretpostavka je da kranji uređaj podržava TCP,UDP protokol.

Pretpostavka je da se korisnik već zna sve informacije vezane za centralni kontroler, kao i da

centralni kontroler isto tako poznaje dovoljno informacija o korisniku tako da mogu da komuniciraju.

## **Reference**

[1] I. Stefanovic: Протокол за комуникацију IP уређаја са централним контролером паметне куће, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2016