**UNIVERZITET U NOVOM SADU**

**FAKULTET TEHNIČKIH NAUKA**

**NOVI SAD**

**Departman za računarstvo i automatiku**

**Odsek za računarsku tehniku i računarske komunikacije**

**ISPITNI RAD**

**Kandidat: Luka Stupar,Stefan Mačak,Strahinja Mršić**

**Broj indeksa: RA67/2017,RA72/2017,RA2/2017**

**Predmet: Osnovi računarskih mreža 2**

**Tema rada: Pametna kuća**

**Mentor rada: Marija Antić,Miloš Pilipović**

**Novi Sad, jun, 2020.**

Sadržaj

[1. Uvod 3](#_Toc43141959)

[2. Realizacija 4](#_Toc43141960)

[3. Održavanje mreže 8](#_Toc43141961)

[4. Upravljanje uređajima (MQTT protokol) 9](#_Toc43141962)

[4.1. Realizacija komunikacionog MQTT modula na strani uređaja 9](#_Toc43141963)

[4.2. Realizacija komunikacionog MQTT modula na strani kontrolera 10](#_Toc43141964)

[4.3. Realizacija konzolne korisničke aplikacije za prikaz stanja uređaja i kontrolu 11](#_Toc43141965)

[5. Grupisanje uređaja i automatska podešavanja 14](#_Toc43141966)

[5.1. Realizacija modula za periodično izvršavanje komandi 14](#_Toc43141967)

[5.2. Realizacija modula za uslovno izvršavanje komandi 14](#_Toc43141968)

[6. Zaključak 15](#_Toc43141969)

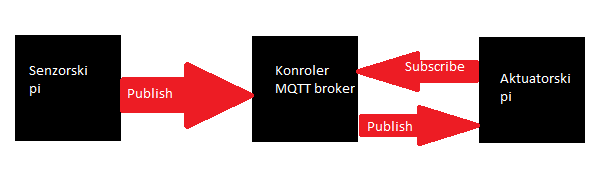
# Uvod

Pametne kuće su kuće koje su kontrolisane sistemom automatizacije. Automatizacija kuće je automatsko centralizovano upravljanje grejanjem, klimatizacijom, ventilacijom, osvetljenjem i ostalim sistemima kuće kroz sistem upravljanja kućom.

Zadatak automatizacije kuće je povećani komfor stanara, smanjena potrošnja energije i duži životni vek instalacija. Stoga je naš cilj da napravimo sistem pametne kuće koji obuhvata sledeće:

* Kontrola temperature
* Kontrola ulaska
* Pametna rerna
* Pametno svetlo

# Realizacija



Senzorski pi prikuplja podatke različitih senzora i objavljuje ih MQTT broker-u koji se nalazi na drugom raspberry pi-u.

MQTT broker omogucuje povezivanje raznorodnih senzora sa aktuatorima. On je smešten na istom raspberrz pi-u kao i kontroler koji vrši kontrolu aktuatora. Broker je smešten na jedan raspberry pi.

Aktuatorski pi prima podatke od Brokerskog , obrađuje ih I određuje sta aktuator treba da odradi ili podatke pretstavlja korisniku.

Teme: Glavna tema je home , ona se deli na hallway, livingroom, bedroom1, beedrom2, kitchen, bathroom.

Hallway ima temu fingerprint.

Livingroom,bathroom,bedroom1 I bedrom2 imaju teme temperature.

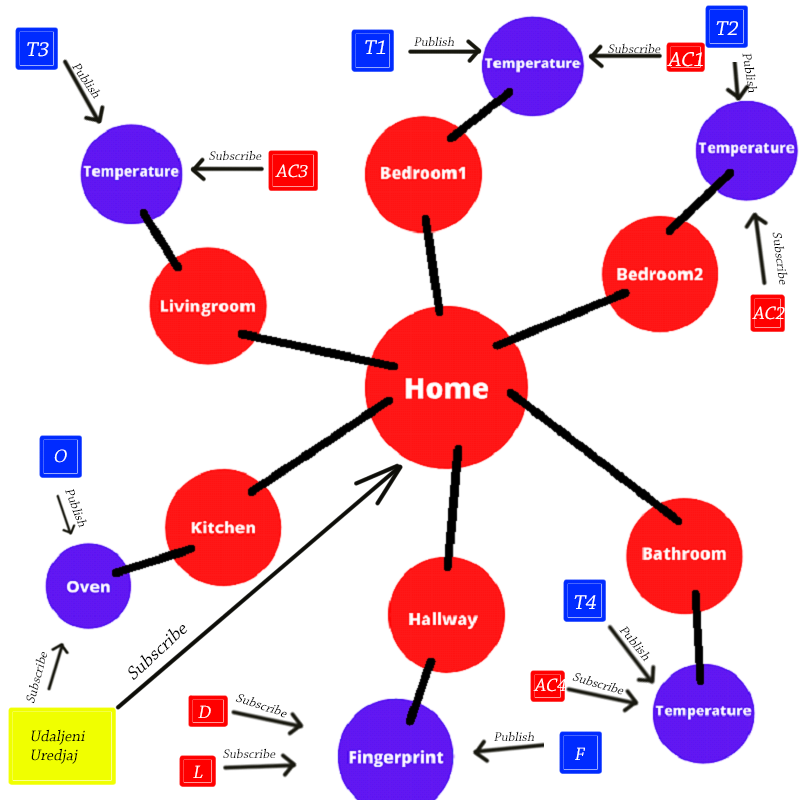
Kitchen ima temu oven.

1.Kontrola temperature:

-Senzor za temperaturu u 4 prostorije

-Akturatori: Klima uredjaj I kontoler grejanja

Senzorski pi na svakih 15 minuta objavljuje informaciju Brokeru u topic: home/\*/temperature.



Aktuatorski pi prima informaciju o temperaturi svake sobe I ako je temperatura manja od 17 stepeni, kontrolerski pi javlja aktuatoru da treba da uključi grejanje a ukoliko je temperatura veća od 25 stepeni uključuje hladjenje. Kada temperatura dodje do optimalnog nivoa (21-22) on isključuje prethodono uključene aktuatore.

2. Kontrola ulaska:

-Senzor otiska prsta

-Aktuator: Servo motoro koji otvara vrata

Ukoliko dođe do aktivnosti na senzoru otiska, pi će obraditi informaciju na osnovu Liste korisnika koji mogu da uđu u kuću. Proslediće Brokeru true ili false na temu home/hallway/fingerprint. Aktuatorski pi prima informaciju I ako je ona true pokrece mehanizam za otvaranje vrata.

3. Pametna rerna:

-Senzor boje na lampici rerne

-Telefon koji prima informaciju

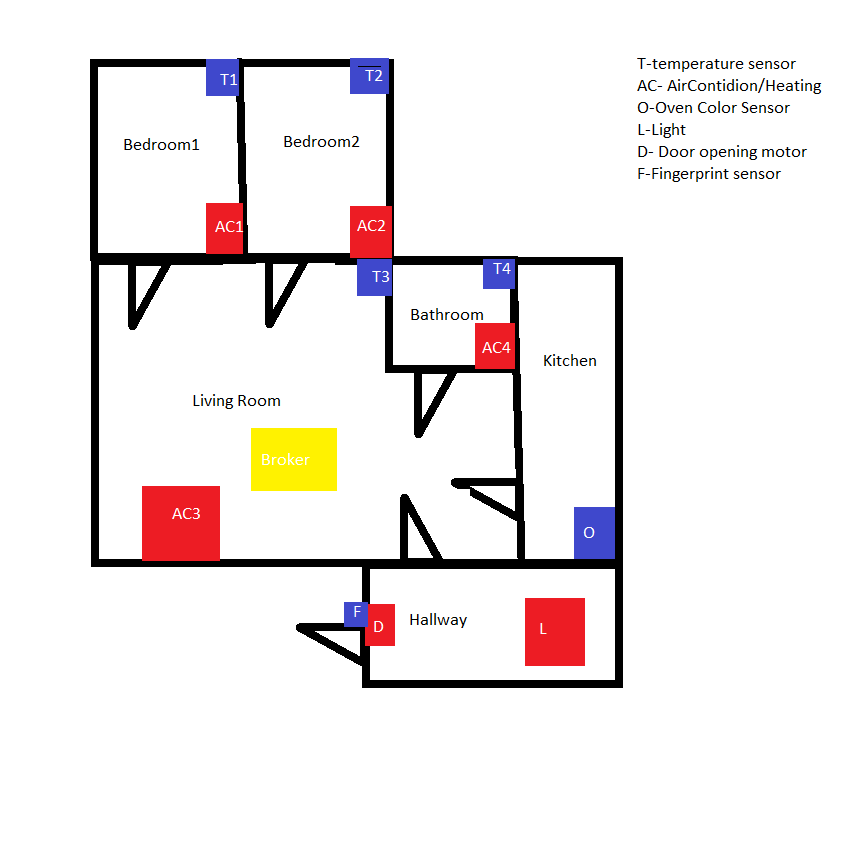
Tema je home/kitchen/oven I tu senzorski pi objavljuje informaciju brokeru ukoliko je rerna uključena na svakih pola sata. To saznaje putem senzora boje na lampici rerne. Pretplatnik je aktuatorski pi koji korisnika obaveštava o tome ako je rerna uključena.

4.Pametno svetlo

-Senzor otiska prsta

-Svetlo

Pi sa kontrolom svetla se pretplacuje na temu fingerprint i svaki put kada dobije potvrdnu informaciju aktiviraće se svetlo(koje ce se ugasiti posle 1 minut.)



# Održavanje mreže

Modul za kontrolu pristupa(server) nam funkcioniše tako što osluškuje broadcast poruke ostalih uređaja i kada dobije poruku od nekog podržanog uređaja njega smešta u evidenciju uređaja na mreži(hash mapa) .Mapa je organizovana da čuva podatke iz klase device.

Klasa device je modelovana:

1. Soba
2. Ime uređaja
3. Podaci o uređaju(i topicima)
4. Klijentska poruka
5. Komunikacioni port
6. Mutex

Koristimo get i set metode za sobe, portove i topike, kao i kreiranje poruke o topicima i podacima.

Uređaj funkcioniše tako što sebi sadrži četiri thread-a. Jedan thread za broadcast, koji šalje poruke za broadcast, drugi koji sluša za poruku servera, treći koji prenosi podatke i četvri odnosno glavni kojivrši interno rukovanje podacima uređaja simulirajući senzore i aktuatore.

Ovaj modul takođe vodi računa o podacima u uređajima koji su na mreži pomoću funkcije update\_device.

Sa druge strane imamo modul za pristupanje mreži sa strane uređaja(device). Sam uređaj je modelovan klasom device koja sadrži ista polja kao I modul za kontrolu pristupa sa razlikom što device potpun konstruktor dok server uzima podatke preko get I set metoda.

Na početku inicijalizuje se sam objekat kojim modelujemo uređaj. Nakon čega se ceo uređaj oglašava preko broadcasta.Nakon poslate broadcast poruke uređaj osluškuje čekajući na odgovor kontrolera (server). Kada device dobije odgovor od kontrolera počinje se sa slanjem podataka o uređaju na dobavljanu adresu servera. Takođe uređaj nastavlja da osluškuje na serverove poruke I vodi računa o tome da li je veza I dalje aktivna se radi preko thread-a.

Ukoliko se desi da dođe do pucanja veze šalje se nova broadcast poruka. Ukoliko server duže vremena ne dobije poruku od određenog uređaja on smatra taj uređaj neaktivnim I osluškuje broadcast port, ukoliko njegova poruka dođe uređaj se smatra aktivnim.

# Upravljanje uređajima (MQTT protokol)

Za relizaciju rešenja korišćen je MQTT-C ,napravljen od strane LiamBandl-a.

Realizovane su sledeće funkcionalnosti:

Get\_Dev\_Info, Set\_Dev\_Info,Get\_Dev\_Value, Set\_Dev\_Value,Get\_Dev\_Info.\* ,Get\_Dev\_Value.\*, --help, Property\_Changed.

Sledi opis rešenja.

## 4.1. Realizacija komunikacionog MQTT modula na strani uređaja

Uredjaji su pretplaćeni na temu sa svojim imenom („Senzorski\_pi“, „Aktuatorski\_pi“). Na toj temi dobijaju poruke od kontrolera a potom na temu „Kontroler“ šalju svoje odgovore.

Svi krajnji uređaji opisani su unutar strukture „**Device**“ koja kao članove ima:

Id – koji pretstavlja jednistveni identifikator uredjaja.

Gropu- grupu kojoj uredjaj pripada, može biti senzor ili aktuator

gpio\_pin – broj pina na koji je krajnji uredjaj povezan

topic – tema na koju senzor objavljuje svoje stanje, a u slučaju aktuataora tema na koju se pretplaćuje

info- sadrži informacije o krajnjem uredjaju

condition- uslov koji se koristi kod aktuatora , na primer kolika je minimalna temperatura koju senzor treba da mu javi da bi on upalio klimu.

**Property\_Changed.Id.Value/Info.Value** – Ovu poruku uredjaj šalje kontroleru kada mu se informacija ili vrednost promeni. Funkcija **update\_sensors\_value** se poziva na svakih 5 sekundi i ukoliko senzor očita različitu vrednost od prethodne na topic „Kontroler“ se objavljuje **property\_changed** poruka. Ova poruka se šalje i kada korisnik zahteva setovanje informacije ili vrednosti.

Funkcija **publish\_callback** prihvata poruku od kontrolera. Zatim je šalje funkcijii **distribute\_pub\_message** koja parsira poruku i u zavisnosti od njenog tipa i šalje je odgovarajućim funkcijama.

1) **Set\_Dev\_Info.Id.Info** – prosledjuje se funkciji **set\_dev\_info** koja prolazi kroz listu krajnjih uredjaja i ukoliko je Id odgovarajći setuje info polje na dato. Potom objavljuje **property\_changed** poruku.

2) **Set\_Dev\_Value.Id.Value** – prosledjuje se funkciji **set\_dev\_value** koja prolazi kroz listu krajnjih uredjaja i ukoliko je Id odgovarajći setuje value polje na dato. Potom objavljuje **property\_changed** poruku.

## 4.2. Realizacija komunikacionog MQTT modula na strani kontrolera

Kontroler je pretplćen na istoimenu temu i tu dobija poruke od strane uredjaja i korisnika preko command line interface-a.

**Publish\_callback** funkcija dobija poruke i proverava da li postoji id koji je napisan u poruci. Ako ne postoji objaviće cli da taj uredjaj ne postoji. Ako postoji proslediće poruku u **distribute\_pub\_message.**

1) **Get\_Device\_Info.Id** – pošto kontroler sadrži bazu podataka sa informacijama o svim uredjajima on će proći kroz niz uredjaja i kada pronadje trazeni objaviće informaciju cli-u.

2) **Get\_Device\_Value.Id** – pošto kontroler sadrži bazu podataka sa informacijama o svim uredjajima on će proći kroz niz uredjaja i kada pronadje trazeni objaviće vrednost cli-u.

3) **Set\_Dev\_Info.Id.Info** – prosledjuje poruku uredjajima i o tome obavestava cli.

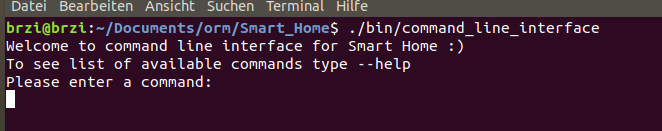
4) **Set\_Dev\_Value.Id.Value** –prosledjuje poruku uredjajima i o tome obavestava cli.

5) **Property\_Changed.Id.Value/Info.Value** – pronaćiće uredjaj u svojoj listi i promeniće mu vrednost ili informaciju.

6)**Get\_Dev\_Info.\*** - spakovaće infomacije o svim uredjajima u poruku i poslati to cli.

7)**Get\_Dev\_Value.\*** - spakovaće infomacije o svim uredjajima u poruku i poslati to cli.

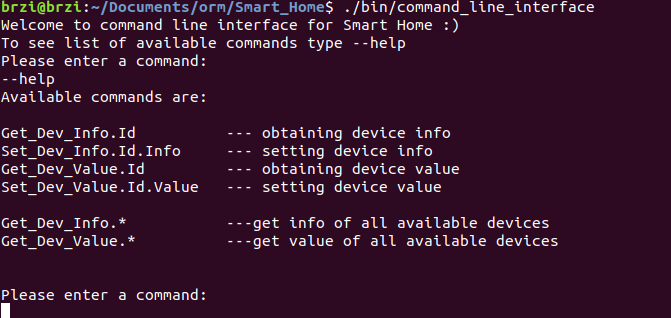
## 4.3. Realizacija konzolne korisničke aplikacije za prikaz stanja uređaja i kontrolu



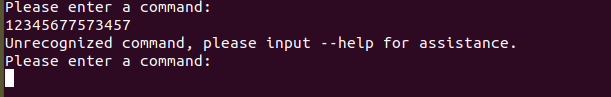
**Command line interface** pretstavlja konzolnu aplikaciju koja korisniku omogućava da dobavi informacije o stanjima uredjaja, i da kontroliše uredjaje( ukoliko su izabrani uredjaji kontrolabilni).

To se postiže kommandama Get\_Dev\_Value, Get\_Dev\_Info, Set\_Dev\_Value i Set\_Dev\_Info.

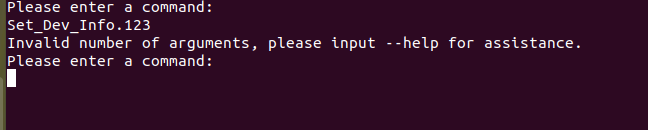
Komanda **–help** omogućuje pregled funkcionalnosti koje korisnik može da koristi.



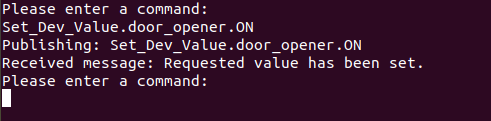
CLI obezbedjuje zaštitu pri unosu nevažećih komandi:



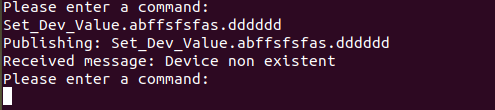
Ukoliko je komanda važeća a format netačan , cli se i o tome stara:



Kada je kommanda validna cli će dobiti potrvrdu od kontrolera da je komanda izvršena:



Ukoliko traženi uredjaj ne postoji kontroler će o tome obavestiti cli na sledeći način:

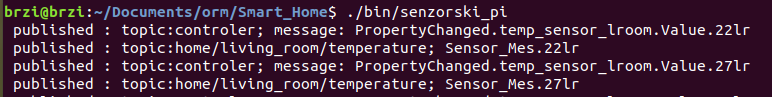


# Grupisanje uređaja i automatska podešavanja

## 5.1. Realizacija modula za periodično izvršavanje komandi

Modul za periodično izvršavanje komandi nalazi se na senzorskom pi-ju.

Na svakih pet sekundi poziva se funkcija **sensors\_value\_publish** koja aktuatorskom pi-ju objavljue informacije o stanju senzora.



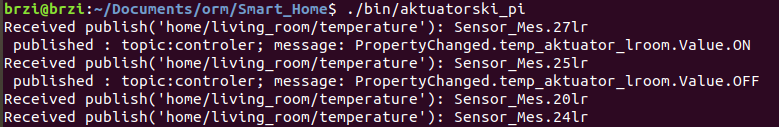
Informacije o stanju senzora dobavljaju se preko funkcije **update\_sensors\_value** koja poziva funkciju **read\_pin**( ova funkcija dobavlja vrednost očitanu sa gpio pina).

## 5.2. Realizacija modula za uslovno izvršavanje komandi

Pod modulom za uslovno izvršavanje podrazumevamo da će aktuator promeniti svoje stanje ako je ispunjen uslov za taj uredjaj( videti kontrolnu tačku 3, član strukture Device je condition).

Ovo se postiže kroz **automation\_control** funkciju koja na osnovu dobijene informacije sa senzora odredjuje da li će menjati stanje aktuatora.

Primer:



U primeru sa slike temp\_aktuator\_lroom ima za uslov 25( ako je temperatura preko 25 stepeni upaliće klimu). Prvi put je temperatura 27 stepeni. Uključuje klimu i obaveštava Kontroler o promeni stanja (Property Changed poruka). Drugi put je temperatura 25 stepeni pa se klima gasi. Nakon toka je 20 pa 24 te oba slučaja ne zahtevaju promenu stanja aktuatora.

# 6. Zaključak

Cilj ovog projekta je sticanje znanja o povezivanju i kontolisanju raznorodnih uređaja putem MQTT protokola. Stekli smo znanje iz projektovanja softvera. Pošto zadatak nije bio predefinisan, nismo želeli da što pre počnemo sa pisanjem koda vec smo prvo izradili detaljan plan projekta kao što bi se radilo u realnoj kompaniji. Rezultat je automatizovana kuća sa većim stepenom energetske efikasnosti, sigurnosti i komfora .