



Diplomski studij

**Informacijska i komunikacijska
tehnologija**

Računarstvo

Telekomunikacije i informatika
Obradba informacija
Računalno inženjerstvo

Ak.g. 2020./2021.

Internet stvari

Upravljanje rasvjetom - Žaruljaši

Projekt

Filip Šošić, 0036508102

Dorian Dončević, 0036510282

Filip Vlašić, 0036510919

Marin Pavić, 0036506982

Marko Maglić, 0036503398

Leon Jukić, 0036501260

Sadržaj

1.....	Uvod
.....	3
2.Opis	rješenja
.....	3
3.IoT	platforma
.....	5
4.Korisničke	aplikacije
.....	8

1. Uvod

IoT sustav u jednoj prostoriji, takav da se ambijent odnosno boja Philips HUE žarulje mijenja s obzirom na trenutačnu temperaturu u prostoriji. Osvjetljenje žarulje pali se i gasi ovisno o prisustvu osobe u prostoriji. Sustav radi na način da senzorom za temperaturu koji je spojen na Wasmote kontroler mjerimo temperaturu prostorije te se ovisno o vrijednosti temperature boja svjetla mijenja. Pomoću MQTT protokola podaci se šalju preko gatewaya na Home Assistant IoT platformu koja se nalazi na virtualnom stroju koji radi u pozadini. Korisnička aplikacija će služiti krajnjem korisniku kako bi manualno mogao mijenjati boju rasvjete te isto tako paliti i gasiti svjetla.

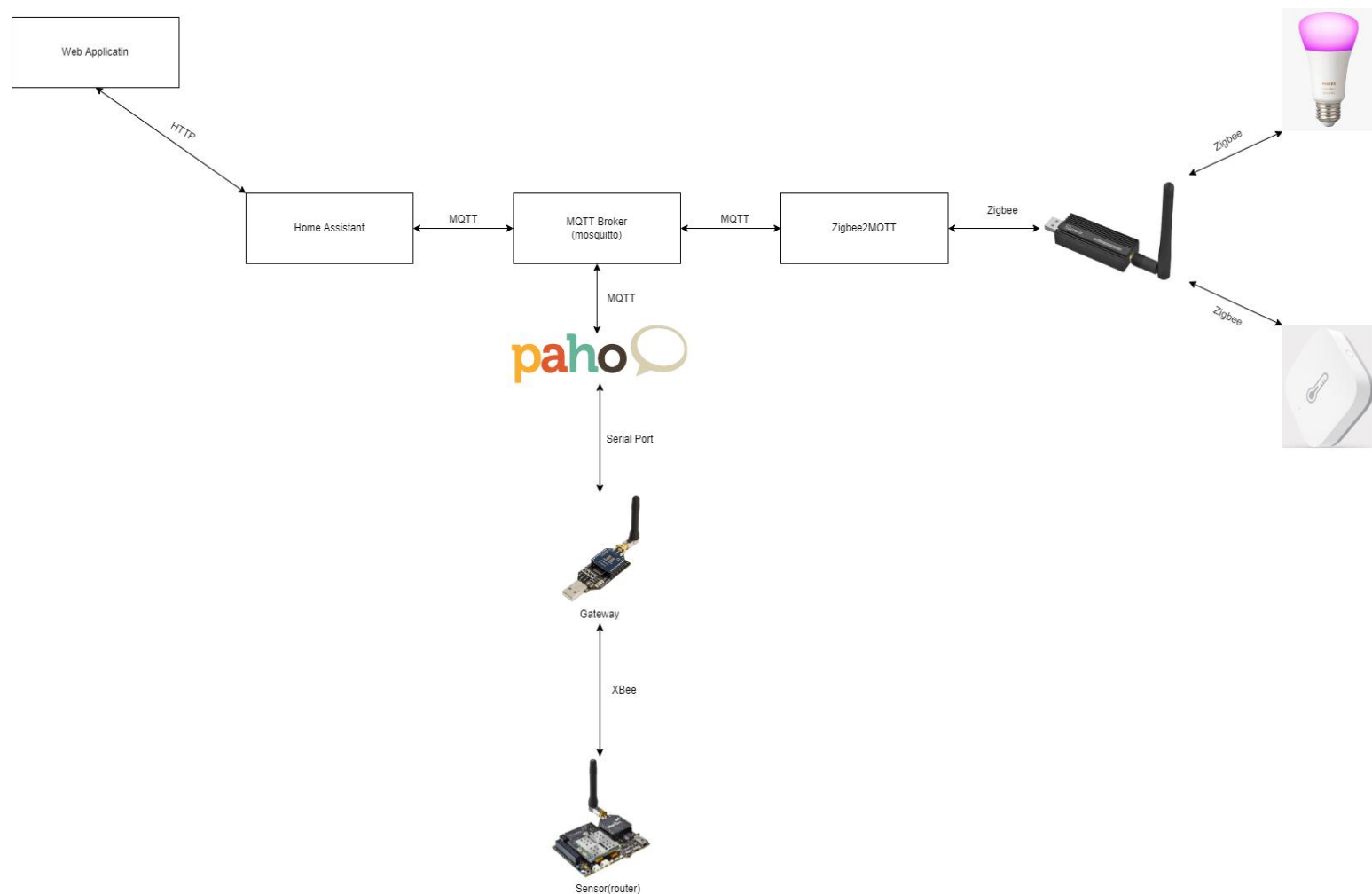
- Koristili smo dva senzora za temperaturu te smo uzimali njihov prosjek (Xiaomi Aquara senzor te obični temperaturni senzor koji je bio spojen na Wasmote uređaj) te smo koristili PIR motion senzor za paljenje s obzirom na prisustvo
- Aktuatori su žarulje koje se pale i gase s obzirom na prisutnost osobe u prostoriji i koje mijenjaju boju te korisnička aplikacija koja manualno može kontrolirati žarulje
- 2 slučaja uporabe:
 - *Primjer: Gašenje osvjetljenja u dnevnom boravku kada u njemu nema nikoga prisutnog radi štednje električne energije*
 - *Primjer: Mijenjanje boje s obzirom na temperaturu unutar prostorije za stvaranje ugodnijeg ambijenta*
- Slični projekti su:
 - Color Temperature Change – *Rambus*
 - Smart Lights – *Smart Home Point*

2. Opis rješenja

- Philips HUE žarulja, Wasmote uređaj, AquaraTemp senzor, temperaturni senzor, PIR motion senzor, Sonoff Zigbee dongle, Wasmote gateway
 - Aktuatori su žarulje te korisnička aplikacija koja direktno komunicira sa Home Assistantom preko http protokola
 - Budući da smo koristili dva različita temperaturna senzora u projektu, koristili smo i dva mrežna prilaza. Jedan od prilaza je bio Xbee gateway preko kojeg je Wasmote senzor slao svoje podatke na MQTT Broker (jedan temperaturni i jedan PIR motion senzor). Drugi prilaz služio je za praćenje informacija od drugog temperaturnog

senzora, te informacija o Philips HUE žarulji, kao i o njezinom upravljanju. Taj prilaz je preko Zigbee protokola komunicirao sa Zigbee2MQTT ekstenzijom podešenom u Home Assistantu kako bi komunicirao sa MQTT Brokerom.

- Kao IoT platformu, kao što smo već naveli, koristili smo Home Assistant koji je bio spojen sa MQTT Brokerom direktno pomoću MQTT protokola. U njemu smo mogli dodavati „virtualne“ senzore sa samog Waspotea, te žarulju i AquaraTemp senzor koji su zatim pratili informacije o uređajima. Isto tako korišten je i za automatizaciju žarulja putem pisanja konfiguracije u .yaml datotekama (točnije configuration.yaml)
 - Korisničke aplikacije koje su korištene u ovom projektu su...(NADOPUNIT)
- Protokol za komunikaciju između aplikacije i platforme je HTTP, komunikacija između platforme i Brokera se provodila putem MQTT protokola kao i komunikacija između Brokera, Xbee gatewaya te Zigbee2MQTT ekstenzije. Za drugi mrežni prilaz koristio se Zigbee protokol prilikom komunikacije između završnih uređaja u arhitekturi i mrežnog prilaza.



Slika 1: Arhitektura rješenja

- Implementacijski detalji
 - Koristili smo dva senzora za temperaturu te smo uzimali njihov prosjek (Xiaomi Aquara senzor te obični temperaturni senzor koji je bio spojen na W aspmote uređaj) te smo koristili PIR motion senzor za paljenje s obzirom na prisustvo
 - Način na koji smo simulirali automatsku akciju aktuatora (žarulja) je mijenjajući temperaturu na sensorima te smo promatrali mijenjanje boja na žaruljama. Druga simulacija bila je pokret ispred PIR senzora kako bi promatrali paljenje i gašenje žarulja. Nakon aktivacije paljenja žarulja od pokreta, postavili smo da se žarulje ugase tek 10 sekundi nakon što nema nikakvog pokreta detektiranog sa strane senzora.
 - Komunikacija između komponenti omogućena je na način da IoT platforma komunicira sa radnim okvirom Mosquitto preko MQTT protokola. Wasmote uređaj je preko mrežnog prilaza Xbee u direktnoj komunikaciji sa radnim okvirom Mosquitto isto tako preko MQTT protokola. Što se tiče prikazivanja mjerenja i akcija na aktuatorima, te akcije se omogućuju preko Zigbee2MQTT radnog okvira. Budući da mrežni prilaz Sonoff koji smo koristili radi na protokolu Zigbee, tu nam je bio potreban taj okvir, kako bi podatke mogli proslijediti na Mosquitto.

3. IoT platforma

- Kao IoT platformu, odlučili smo se na korištenje Home Assistant-a.
- Home Assistant je besplatna IoT platforma koja služi za podešavanje i kontroliranje IoT uređaja. Home Assistant je međutim platforma koja mora biti osposobljena na vlastitom serveru, tako da imamo opcije za postavljanje platforme na virtualnom računalu, pomoću Dockera ili pak na Raspberry Pi-u. Mi smo se odlučili postaviti Home Assistant na virtualno računalo, spojiti ga na LAN te mu pristupati preko lokalne IP adrese.
- Home Assistant relativno je jednostavan za korištenje pogotovo ako koristimo uređaje koji su kompatibilni s njim. Također jako je koristan i u ostvarivanju drugih komunikacija zbog mogućnosti instalacije radnih okvira.
- Budući da smo koristili Wasmote uređaj i xbee mrežni prilaz, važno nam je bilo instalirati 2 okvira koja su nam bila potrebna u komunikaciji. To su Mosquitto MQTT radni okvir (broker) i Zigbee2MQTT za drugi mrežni prilaz. To nam je omogućilo komunikaciju među IoT sensorima i aktuatorima te postavljanje njihove konfiguracije.
- U kontrolnoj ploči na Home Assistantu dodali smo Visual Studio Code editor pomoću kojega smo zatim čitali stanja fizičkih senzora te kontrolirali Philips HUE žarulje. Kod se pisao u

config.yml datoteci. Ovdje nam je isto tako bilo važno stvaranje trećeg virtualnog senzora koji nam je služio za prikaz prosječne temperature s mjerenja naša dva fizička senzora.

- Na kraju, stanja svih tih senzora te stanje samih žarulja može se lako pregledati preko kontrolne ploče u Home Assistant-u. Stanja žarulja tamo možemo ručno postavljati, te mijenjati boje.

```
19 automation on_temp_change_turn_light_on:
20   alias: On temperature change, change bulb color
21   trigger:
22     platform: state
23     entity_id: sensor.average_temperature
24   condition:
25     condition: state
26     entity_id: binary_sensor.waspmotepir
27     state: "on"
28   action:
29     - entity_id: light.philipsbulb
30       service: light.turn_on
31       data:
32         hs_color:
33           - >
34             {% set aqaraTemp = states('sensor.average_temperature') | float%}
35             {% if aqaraTemp < 20 %}
36               240
37             {% elif (aqaraTemp >= 20 and aqaraTemp < 21) %}
38               220
39             {% elif (aqaraTemp >= 21 and aqaraTemp < 22) %}
40               200
41             {% elif (aqaraTemp >= 22 and aqaraTemp < 23) %}
42               190
43             {% elif (aqaraTemp >= 23 and aqaraTemp < 24) %}
44               170
45             {% elif (aqaraTemp >= 24 and aqaraTemp < 25) %}
46               100
47             {% elif (aqaraTemp >= 25 and aqaraTemp < 26) %}
48               80
49             {% elif (aqaraTemp >= 26 and aqaraTemp < 27) %}
50               60
51             {% elif (aqaraTemp >= 27 and aqaraTemp < 28) %}
52               40
53             {% elif (aqaraTemp >= 28 and aqaraTemp < 29) %}
54               20
55             {% elif (aqaraTemp >= 29 and aqaraTemp < 30) %}
56               10
57             {% else %}
58               0
59             {% endif %}
60           - 100
```

Slika 2: Konfiguracija automatizacije mijenjanje boje svjetla.

```
automation on_no_motion_turn_off_light:
```

```
  - alias: "On no motion turn off light"
  - trigger:
    - platform: state
    - entity_id: binary_sensor.waspmotepir
    - to: "off"
    - for:
      - seconds: 10
  - action:
    - service: light.turn_off
      target:
        entity_id:
          - light.philipsbulb
          - light.philipsbulb2
          - light.philipsbulb3
```

```
binary_sensor:
```

```
  - platform: mqtt
    name: "WaspMotePir"
    state_topic: "zigbee2mqtt/pirWaspMote"
    payload_on: "1"
    payload_off: "0"
    value_template: "{{value_json.state}}"
```

Slika 3: Konfiguracija automatizacije za senzor, te konfiguracija samog senzora sa WaspMotea

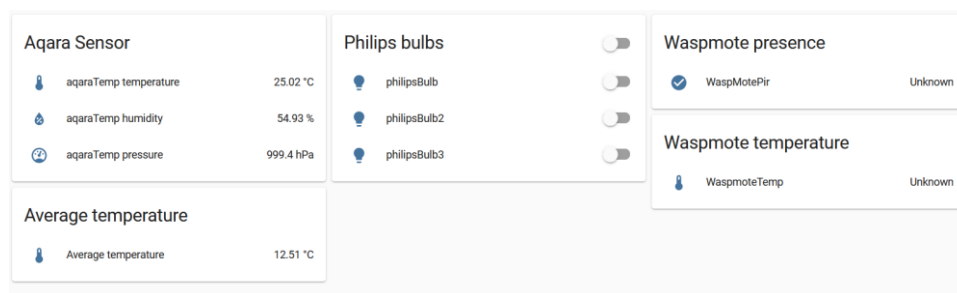
```
sensor:
```

```
  - platform: mqtt
    state_topic: zigbee2mqtt/tempWaspMote
    name: "WaspMoteTemp"
    unit_of_measurement: "°C"
    value_template: "{{value_json.state}}"
```

```
template:
```

```
  - sensor:
    - name: "Average temperature"
      unit_of_measurement: "°C"
      state: >
        {%set tempAqara = states('sensor.aqaratemp_temperature') | float%}
        {%set tempWaspMote = states('sensor.waspmotetemp') | float%}
        {% set avgTemp = ((tempAqara + tempWaspMote) / 2) | round(2) %}
        {{avgTemp}}
```

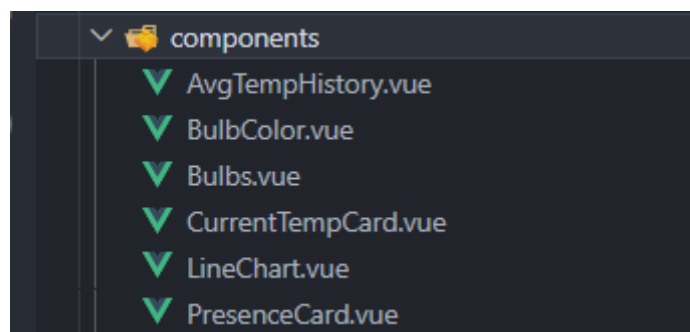
Slika 4: Konfiguracija temperaturnog senzora sa WaspMote uređaja, te stvaranje novog, virtualnog senzora koji uzima prosjek dva temperaturna senzora.



Slika 5: Prikaz podataka na kontrolnoj ploči na Home Assistant-u.

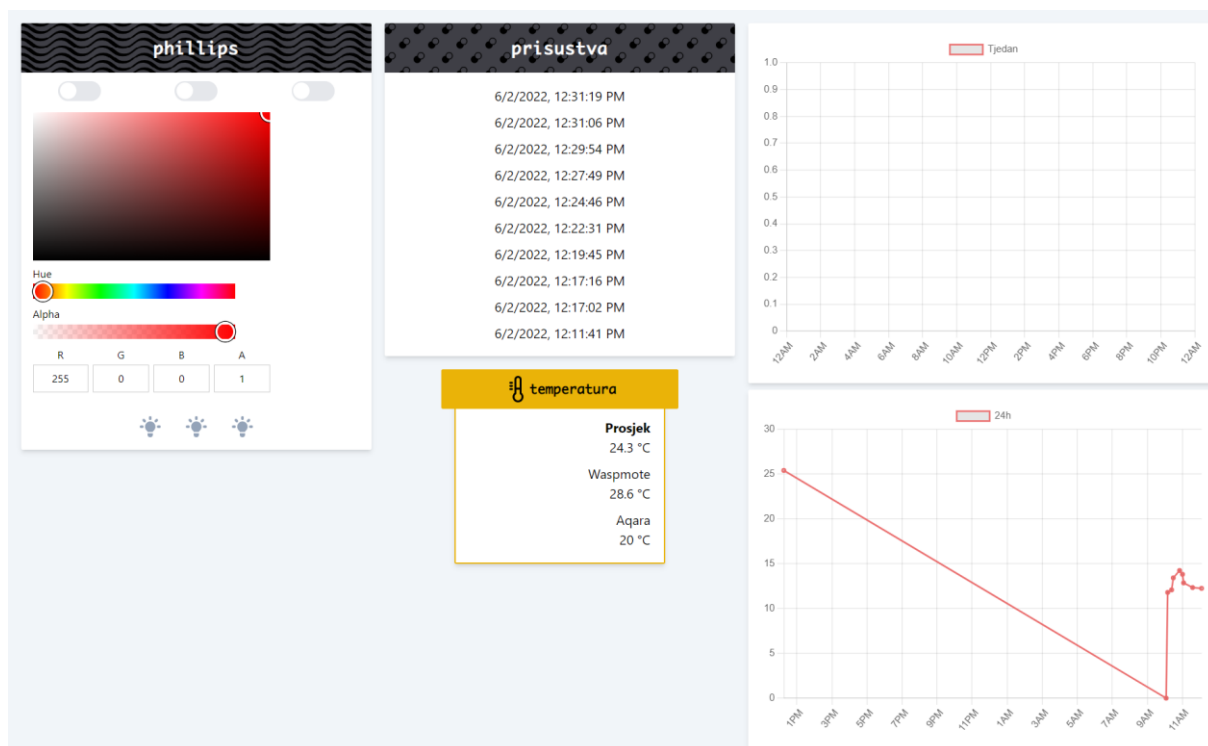
4. Korisničke aplikacije

Korisnička web-aplikacija je napravljena kombinacijom **Vue.js** i **Tailwind** radnih okvira. Vue je JavaScript okvir za izgradnju korisničkih sučelja i *single-page* aplikacija. Tailwind je CSS (eng. *Cascading Style Sheets*) okvir za brzu i jednostavnu izgradnju korisničkih sučelja i kompatibilan je s Vue-om. Vue.js se oslanja na hijerarhijsku strukturu komponenti koje međusobno komuniciraju. Sve izrađene komponente se nalaze u istom pogledu (engl. View).



Slika 4-1 Popis komponenti aplikacije

Krajnji produkt je web aplikacija koja prikazuje trenutno stanje svih temperaturnih senzora i svjetiljki. Većina komponenti tijekom stvaranja šalje zahtjev prema poslužitelju Home-Assistant za svoje podatke. Uočavaju se komponente za prikaz prisustva (*PresenceCard.vue*), upravljanje Philips žaruljama (*Bulbs.vue*, *BulbColor.vue*) te grafovi koji prikazuju temperaturu u proteklih 24 sata i tjedan dana. Posljednja komponenta je komponenta za trenutnu temperaturu koja prikazuje aktualno stanje temperature te se osvježava svaku sekundu.



Slika 4-2 Konačni izgled aplikacije