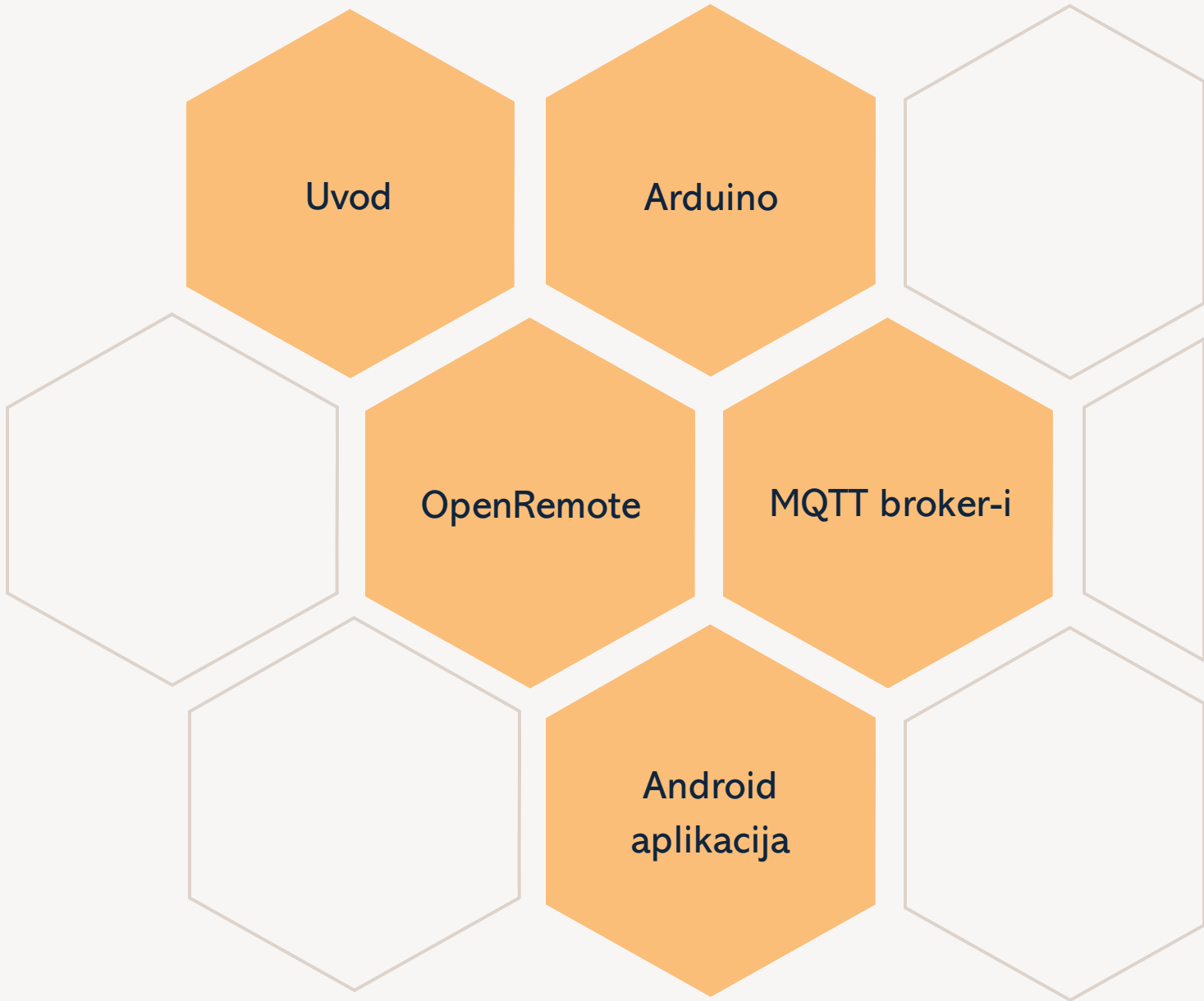
Transportni Informacioni Sistemi

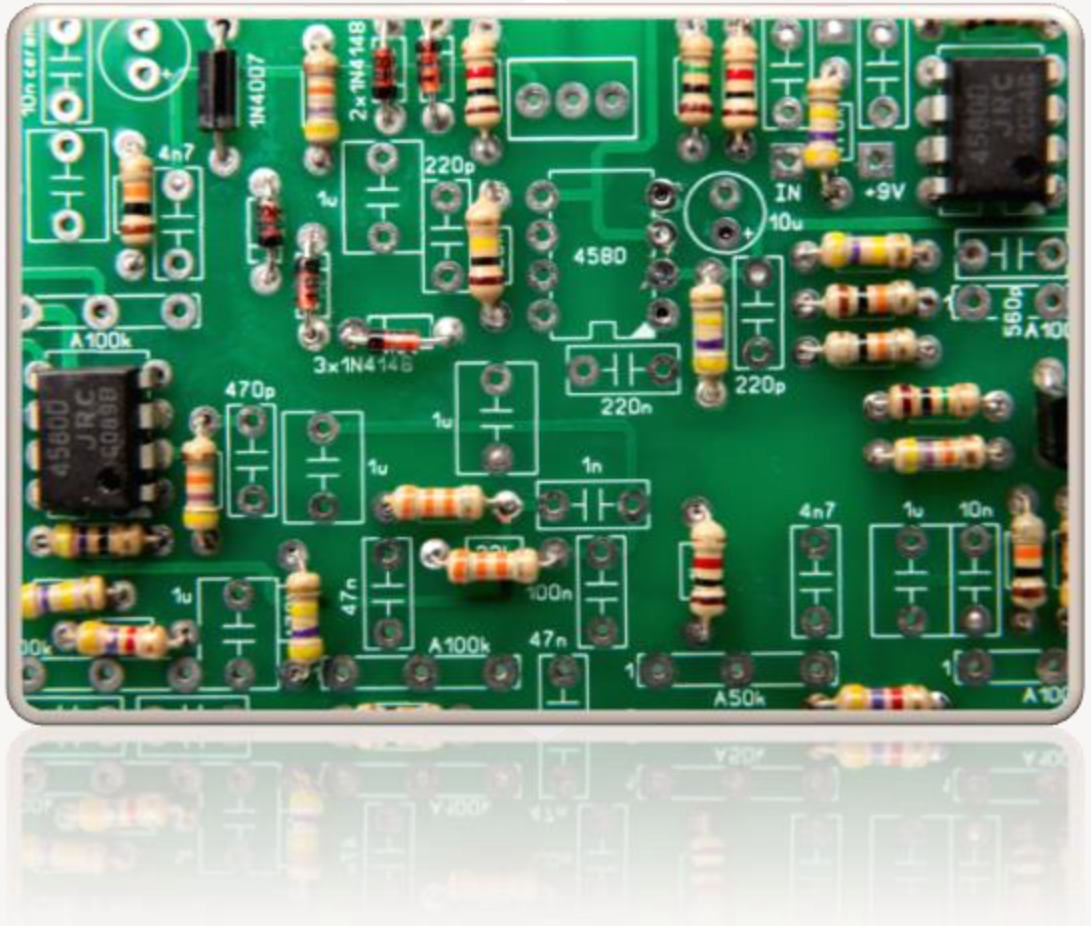
Student:
Teodora Kocić, 1457

Sadržaj



Uvod

SmartCar sistem se sastoji iz servisa koji prikuplja podatke sa senzora Arduino-a i šalje ih putem MQTT broker-a na edge - OpenRemote platforma. Na edge-u pristigli podaci se filtriraju i obrađuju primenom napisanih pravila. Obrađeni podaci šalju se putem Mosquitto broker-a mobilnoj aplikaciji, gde korisnik u realnom vremenu može da prati spoljne parametre i promenu istih. Kao odgovor na date promene izvršava se neki tip akcije na aktuatoru (Arduino).


1N4007
2x1N4148
4n7
220p
1u
IN
10u
+9V
4580
A100k
3x1N4148
470p
220n
1n
4n7
10n
100n
47n
A50k

Arduino

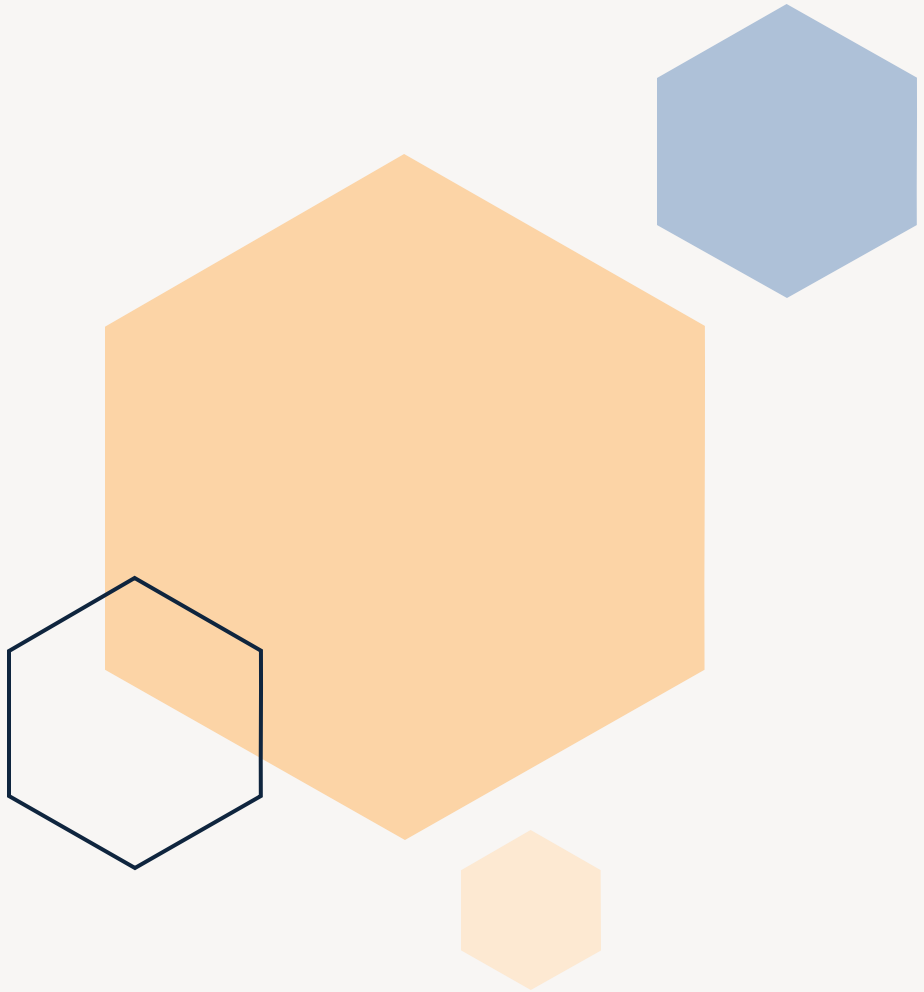
```
void setup() {
  Serial.begin(9600);
  while (!Serial);

  pinMode(ledPin, OUTPUT); // use the LED as an output
  pinMode(buttonPin, INPUT); // use button pin as an input

  // begin initialization
  if (!BLE.begin()) {
//    Serial.println("starting Bluetooth® Low Energy module failed!");

    while (1);
  }

  BLE.setLocalName("NANO 33 BLE");
  BLE.setAdvertisedService(ledService);

  ledService.addCharacteristic(ledCharacteristic);
  ledService.addCharacteristic(buttonCharacteristic);

  BLE.addService(ledService);

  ledCharacteristic.writeValue(0);
  buttonCharacteristic.writeValue(0);

  BLE.advertise();
//  Serial.println("Bluetooth® device active, waiting for connections...");

  if (!BARO.begin()) {
    Serial.println("Failed to start the LPS22HB sensor");
    while (1);
  }

  if (!APDS.begin()) {
    Serial.println("Failed to start the APDS9960 sensor");
    while (1);
  }

  if (!IMU.begin()) {
    Serial.println("Failed to start the LSM9DS sensor.");
    while (1);
  }
}

void loop() {
  BLE.poll();

  float pressure = BARO.readPressure(); // In kPa
  float temp = BARO.readTemperature();

//  int gesture = APDS.readGesture();
  int proximity = APDS.readProximity();

  int r, g, b, a;
  APDS.readColor(r, g, b, a);
```

Korišćen je Arduino Board Nano 33 BLE Sense. Servis koji je napisan u Python-u čita podatke sa serijskog porta i šalje ih na odgovarajuće topic-e MQTT broker-a.

Takođe Arduino se koristi za simulacija aktuatora koji izvršava komande (BLE veza) kada određeni parametri pređu definisane vrednosti granica. LED dioda na Arduino-u, u zavisnosti od slučaja korišćenja, treperi u različitim bojama (žuto, plavo, zeleno i crveno).

Arduino senzori

Sensor Name	Type 1	Type 2	Type 3
LPS22HB	Temperatura	Atmosferski pritisak	--
APDS9960	Boja (RGB + Alpha)	Pokret	Udaljenost
LSM9DS1	Akcelerometar	Žiroskop	Magnetometar

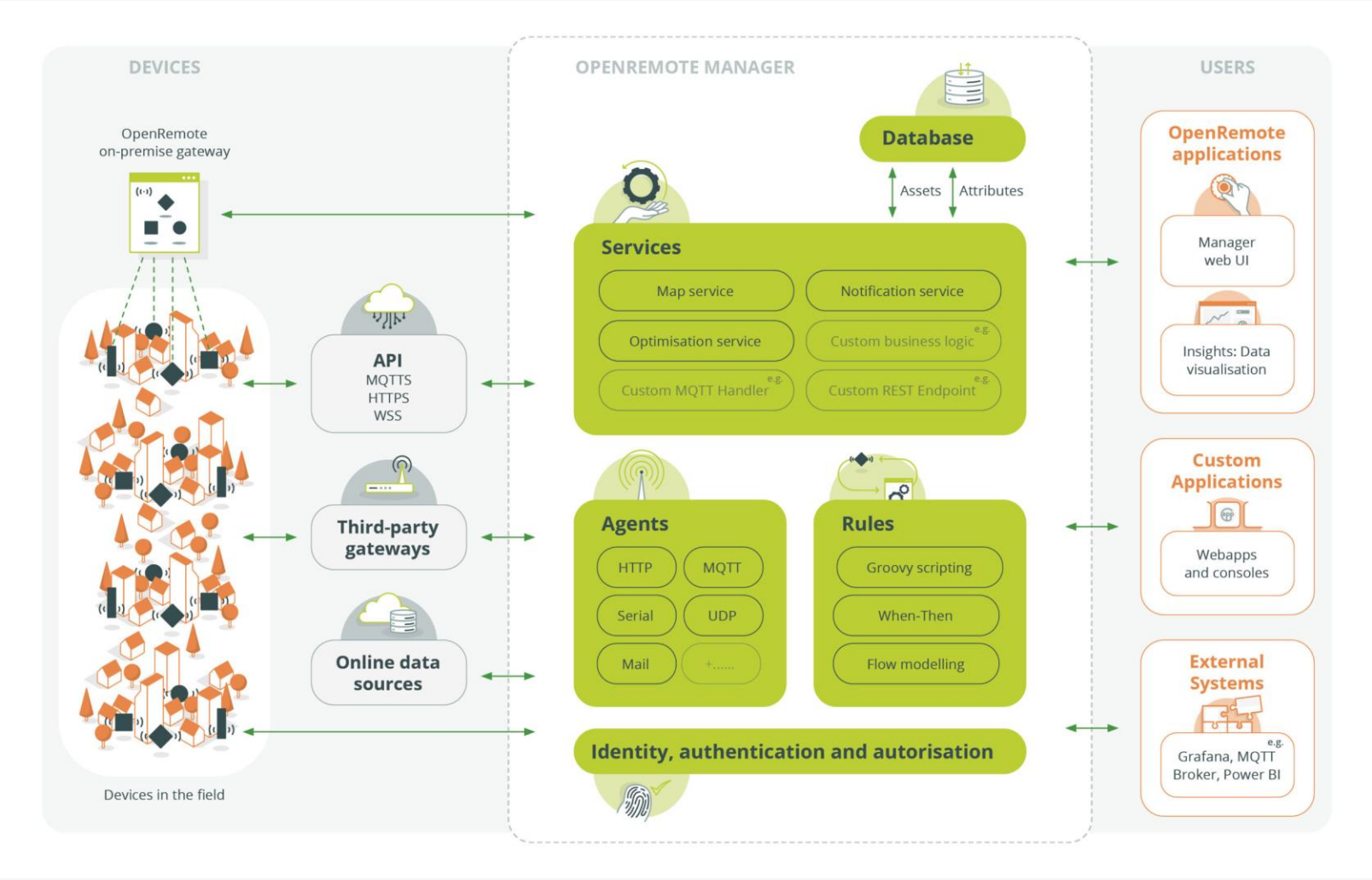


Arduino kroz korisnički kreirane servise i karakteristike prima komande od mobilne aplikacije.
U zavisnosti od vrednosti koja se šalje kroz komande, LED dioda Arduino-a treperi plavo ukoliko je spoljno osvetljenje slabo, zeleno kada su parking senzori aktivni i crveno kada se uključuje sistem za stabilnost (sistem kočenja).
Dioda najpre treperi žuto kada se uspešno izvrši konekcija mobilnog uređaja sa Arduino baord-om.

OpenRemote

- **100% open-source IoT platforma**
- **Jednostavno povezivanje mrežnih sredstava (assets) sa mobilnim/web aplikacijama**
- ***Manager* je aplikacija napisana u Java-i koja predstavlja srž sistema. Formira broker IoT konteksta i tako beleži trenutno stanje sistema**
- **Sigurnost IoT sistema bazira se na industrijskom standardu za autentifikaciju – integracija *Manager*-a i *Keycloak*-a**
- **Skladištenje podataka u *PostgreSQL* bazi podataka**

Arhitektura platforme



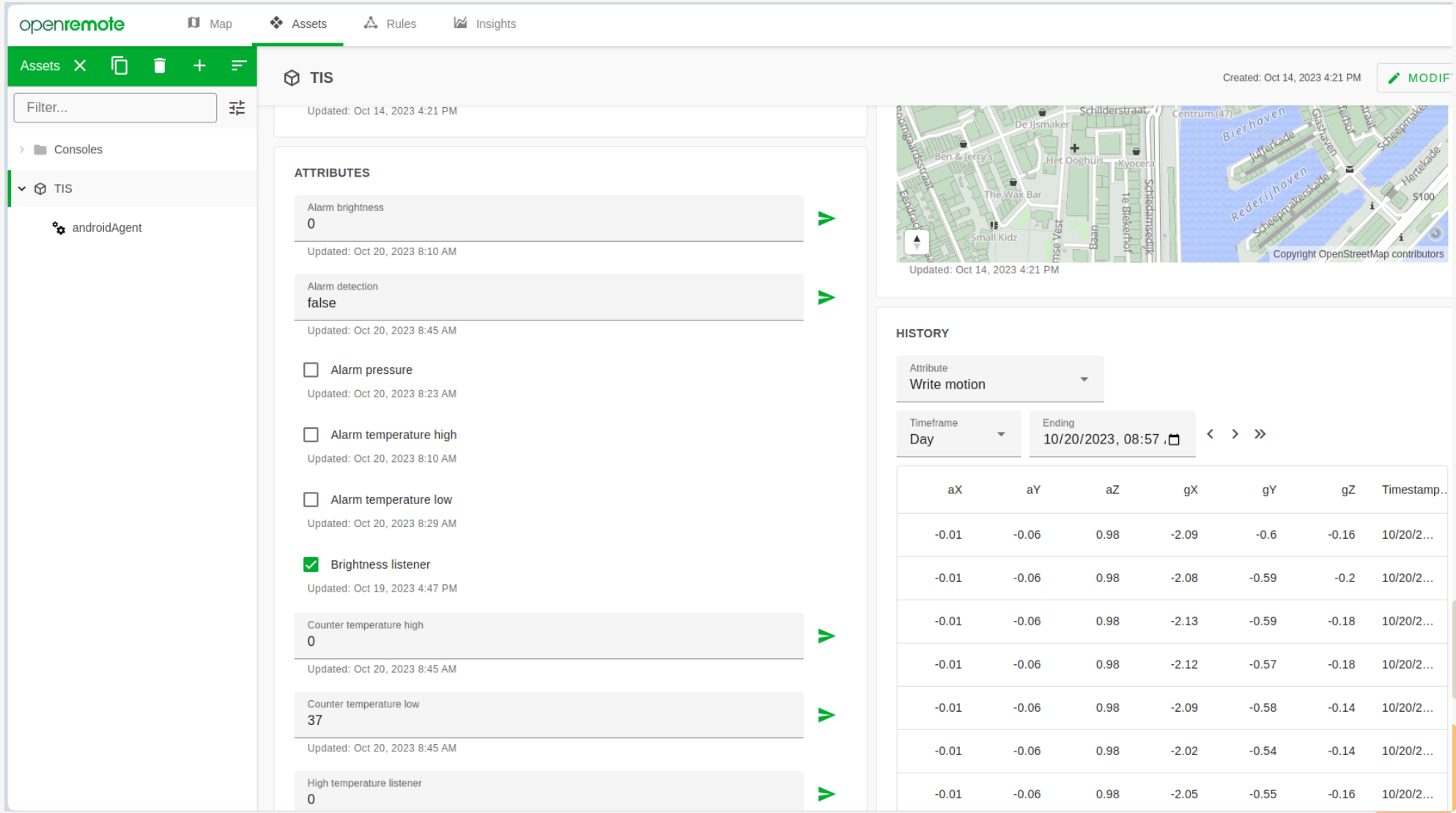
OpenRemote-ov *docker compose* fajl

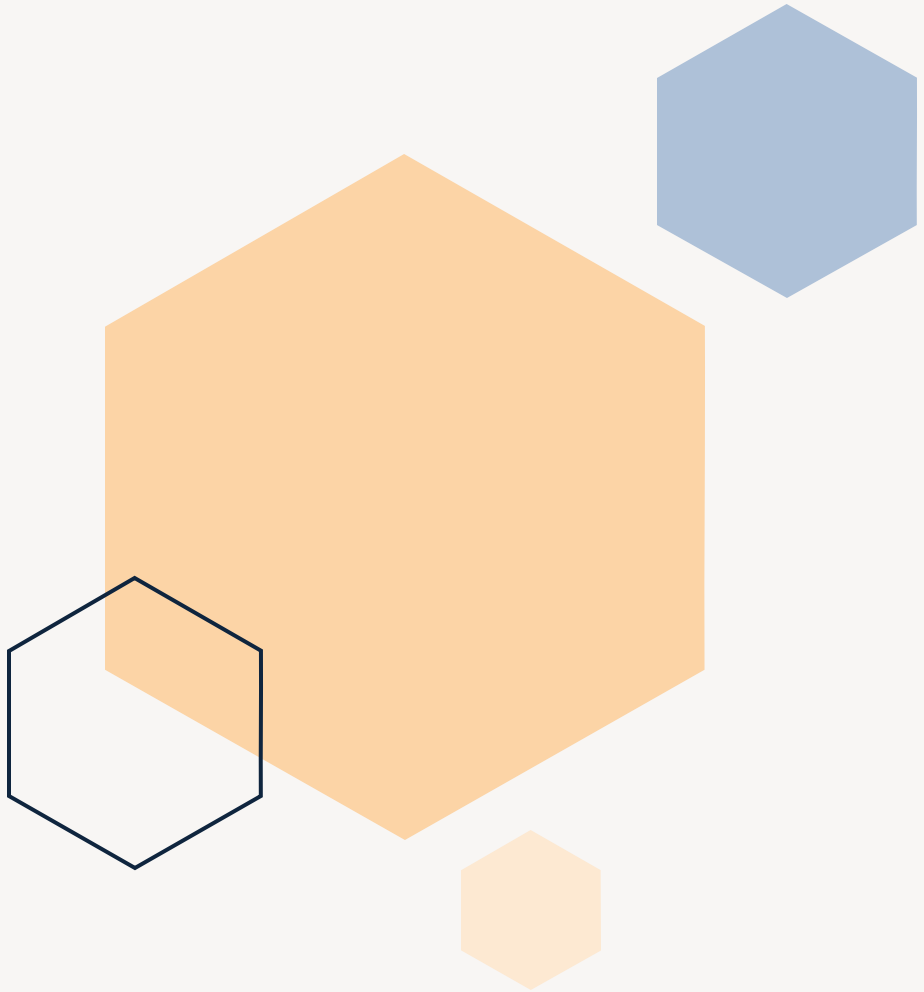
```yaml
volumes:
  proxy-data:
  manager-data:
  postgresql-data:

services:

  proxy:
    image: openremote/proxy:${PROXY_VERSION:-latest}
    restart: always
    depends_on:
      manager:
        condition: service_healthy
    ports:
      - "80:80" # Needed for SSL generation using letsencrypt
      - "${OR_SSL_PORT:-443}:443"
      - "8883:8883"
      #- "127.0.0.1:8404:8404" # Localhost metrics access
    volumes:
      - proxy-data:/deployment
    environment:
      LE_EMAIL: ${OR_EMAIL_ADMIN:-}
      DOMAINNAME: ${OR_HOSTNAME:-localhost}
      DOMAINNAMES: ${OR_ADDITIONAL_HOSTNAMES:-}
      # USE A CUSTOM PROXY CONFIG - COPY FROM https://raw.githubusercontent.com/c
      #HAPROXY_CONFIG: '/data/proxy/haproxy.cfg'

  postgresql:
    restart: always
    image: openremote/postgresql:${POSTGRESQL_VERSION:-latest}
    volumes:
      - postgresql-data:/var/lib/postgresql/data
      - manager-data:/storage

  keycloak:
    restart: always
    image: openremote/keycloak:${KEYCLOAK_VERSION:-latest}
    depends_on:
      postgresql:
        condition: service_healthy
    volumes:
      - ./deployment:/deployment
    environment:
      KEYCLOAK_ADMIN_PASSWORD: ${OR_ADMIN_PASSWORD:-secret}
      KC_HOSTNAME: ${OR_HOSTNAME:-localhost}
      KC_HOSTNAME_PORT: ${OR_SSL_PORT:--1}


  manager:
#    privileged: true
    restart: always
    image: openremote/manager:${MANAGER_VERSION:-latest}
    depends_on:
      keycloak:
        condition: service_healthy
    ports:
      - "1883:1883"
    #  - "127.0.0.1:8405:8404" # Localhost metrics access
    environment:
      OR_SETUP_TYPE:
      OR_ADMIN_PASSWORD:
      OR_SETUP_RUN_ON_RESTART:
      OR_EMAIL_HOST:
      OR_EMAIL_USER:
      OR_EMAIL_PASSWORD:
      OR_EMAIL_X_HEADERS:
      OR_EMAIL_FROM:
      OR_EMAIL_ADMIN:
      OR_HOSTNAME: ${OR_HOSTNAME:-localhost}
      OR_ADDITIONAL_HOSTNAMES: ${OR_ADDITIONAL_HOSTNAMES:-}
      OR_SSL_PORT: ${OR_SSL_PORT:--1}
```

Komponente OpenRemote platforme

Assets (sredstva)	Agents (agenti)	Rules (pravila)	Insights (pregled)
Skup svih atributa za koje postoji opis i istorija promena. Takođe, moguće je dodavati nove atribute i kreirati konfiguraciju za iste	Specifična vrsta sredstava čija je namena da omoguće povezivanje sa eksternim senzorima, uređajima, aktuatorima, itd. (MQTT, Web sockets, HTTP, TCP, UDP, Z-Wave)	Postoje tri tipa pravila: ***when-then*** (pokreću akciju nad atributom koja prethodi promeni nekog drugog atributa), ***flow*** (obrađuju atribute i potom kreiraju nove) i ***groovy*** (atributi se upravljaju naprednom logikom)	Koristi se za kreiranje dashboard-ova u okviru manager-a

Sredstva, agenti i atributi u projektu SmartCar





TIS

Updated: Oct 20, 2023 8:29 AM

Write alpha
38
Updated: Oct 20, 2023 8:45 AM

Write blue
10
Updated: Oct 20, 2023 8:45 AM

Write green
12
Updated: Oct 20, 2023 8:45 AM

Write motion
{
"aX": -0.01,
"aY": -0.06,
"aZ": 0.98,
"gX": -2.09,
Updated: Oct 20, 2023 8:45 AM

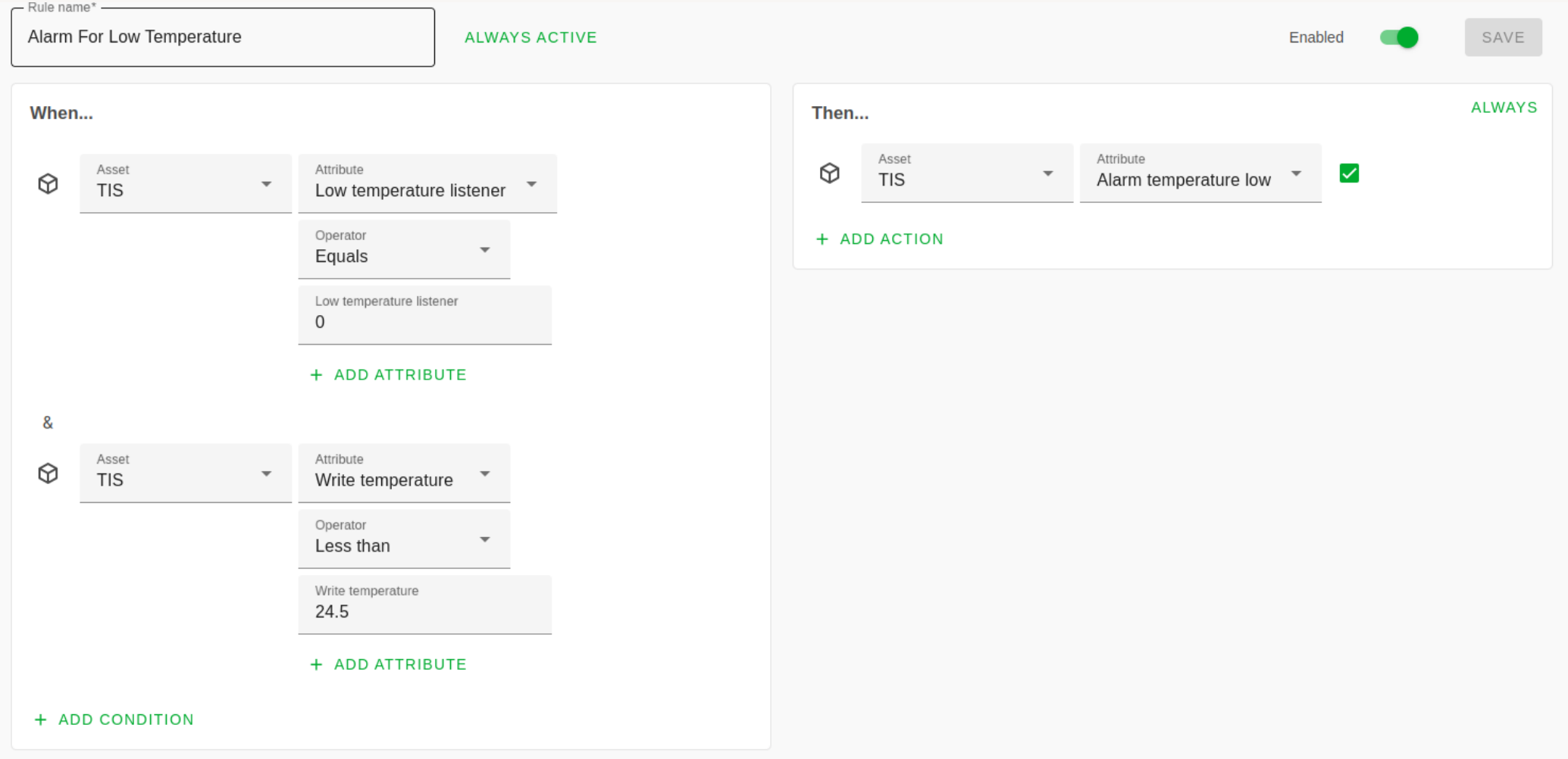
Write pressure
98.61
Updated: Oct 20, 2023 8:45 AM

Write proximity
200
Updated: Oct 20, 2023 8:45 AM

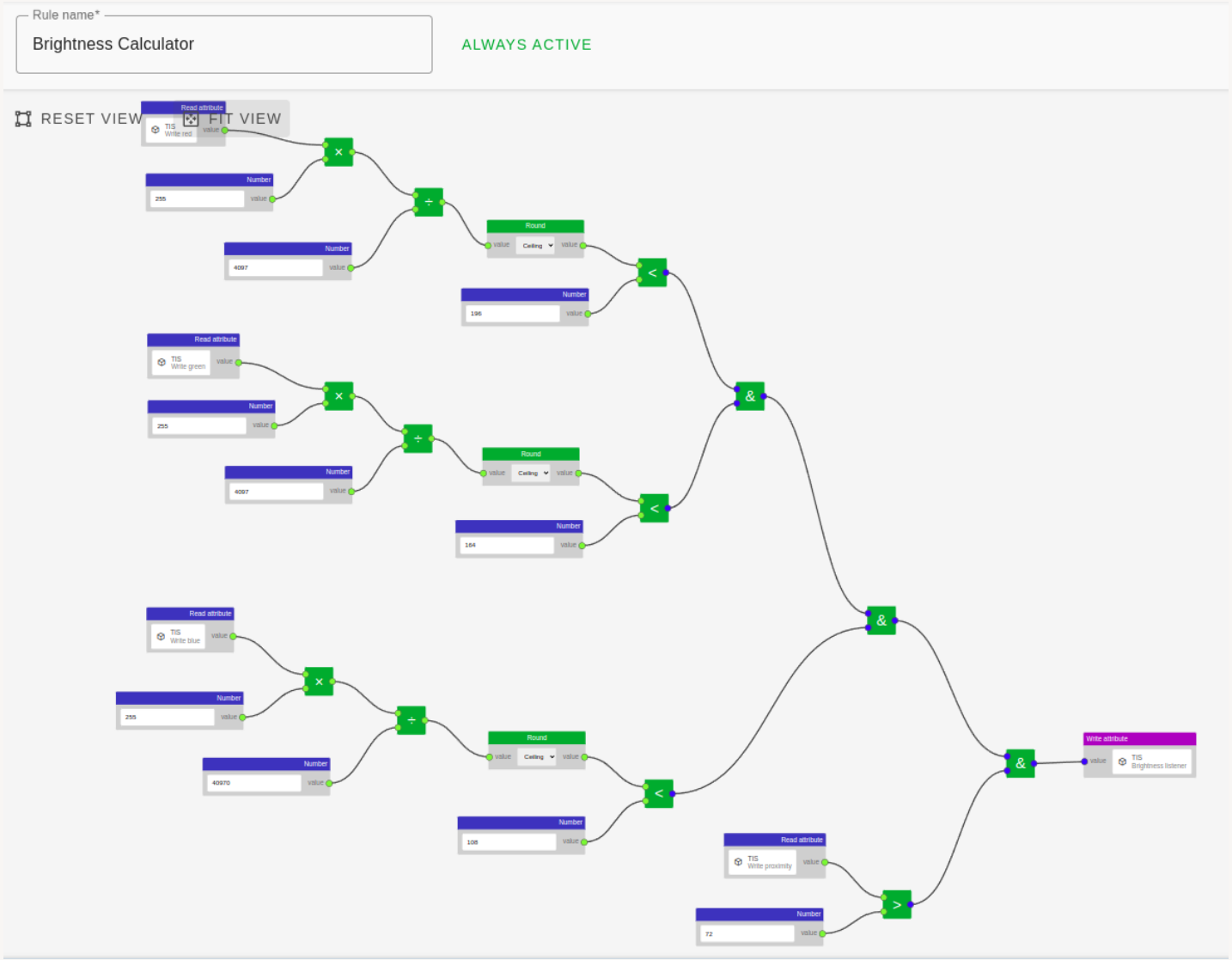
Write red
25
Updated: Oct 20, 2023 8:45 AM

Write temperature
28.32

When-Then pravilo za aktivaciju klime u vozilu kada temperatura padne ispod donje granice

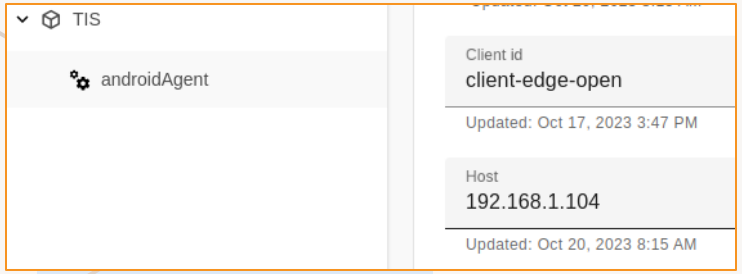


Flow pravilo za računanje spoljnog osvetljenja



Dashboard za linearno i rotaciono ubrzanje

MQTT broker-i



OpenRemote-ov default-ni broker

- Host je adresa **proxy** kontejnera
- Sluša na port-u **8883**, odnosno **1883** za slučaj bez TLS verifikacije
- Na kreirane **topic**-e (korišćenjem OpenRemote *Manager*-a) šalju se sample-ovani podaci iz Python script-e koja dobija vrednosti senzorskih podataka sa serijskog port-a Arduino-a

Mosquitto broker

- Host je na adresi Wi-Fi mreže na koju je uređaj **konektovan**
- Sluša na port-u **1884**
- Mqtt agent (kreiran pomoću OpenRemote *Manager*-a) filtrirane podatke sa edge-a šalje na nove Mosquitto topic-e
- Na pomenute topic-e pretplaćena je Android mobilna aplikacija koja prikazuje te podatke

Komunikacija preko topic-a

Broker OpenRemote platforme

- Podaci se šalju na topic-e:

  {realm}/{clientId}/writeatrributevalue/{attributeName}/{assetId},

gde su vrednosti *realm* i *clientId* vrednosti koje korisnik kreira direktno na svom profilu u okviru OpenRemote Manager-a, *attributeName* naziv posmatranog atributa, dok se vrednost *assetId* isčitava direktno iz URI-a kada se korisnik nalazi na web stranici sredstava u okviru OpenRemote Manager UI-a

Mosquitto broker

- Podaci se šalju tako što se atribut sa nazivom *attributeName* poveže sa Mqtt agentom Mosquitto broker-a i potom kreira novi topic:

  edge/{subscribeAttributeName},

gde je *subscribeAttributeName* deo filtera u Android aplikaciji kojim se definiše precizno koji podatak je stigao

- Mqtt klijent u okviru Android aplikacije osluškuje na:

  edge/#

Android aplikacija

Home Screen

Stanje automobila (motor, vrata, klima, svetla) i čitanje podataka iz lokalne baze

Controls Screen

Prikaz temperature i pritiska. Po potrebi paljenje klime i ventilacije

Stability Screen

Prati kretanje automobila i po potrebi aktivira kočioni sistem. Kontrola svetala, kao i parking senzora

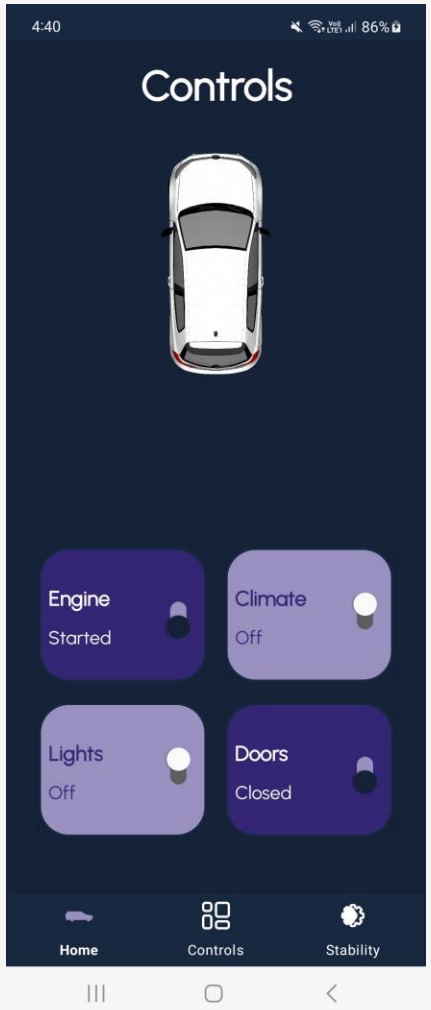
Mqtt klijent

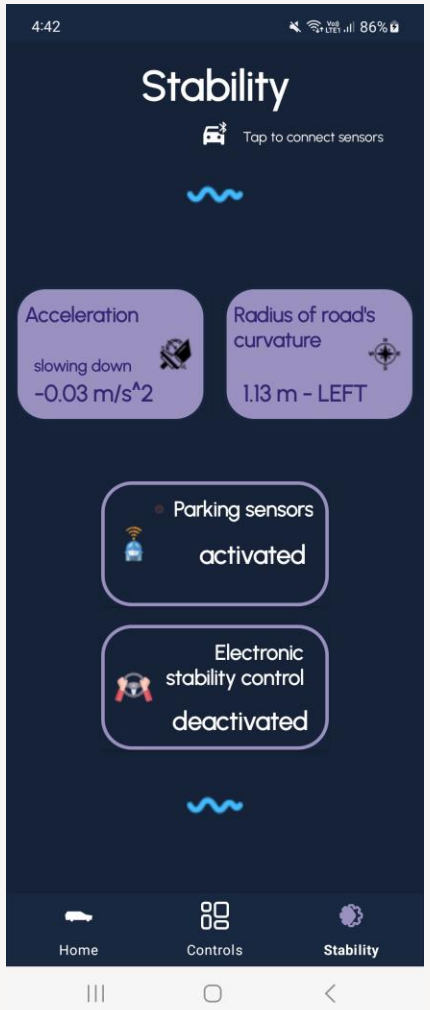
Za očitavanje obrađenih i filtriranih podataka sa edge-a (OpenRemote platforme)

BLE konekcija

Omogućava pregled predefinisanih i korisnički kreiranih karakteristika i servisa. Slanje komandi Arduino-u

Slike ekrana SmartCar aplikacije

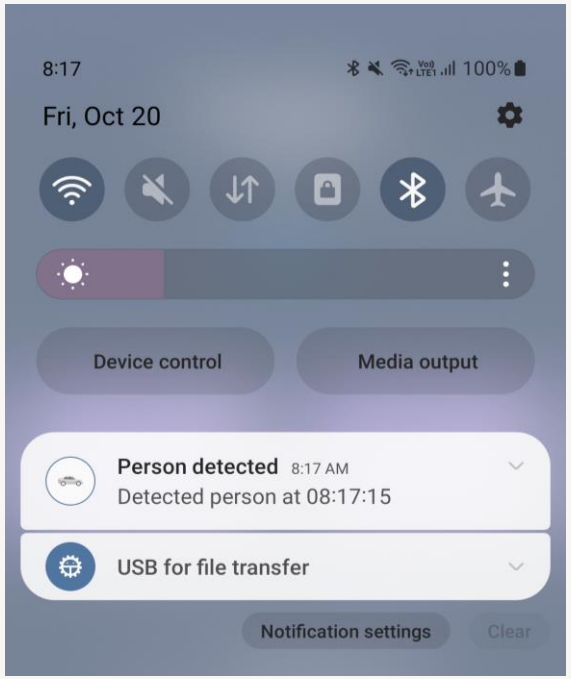


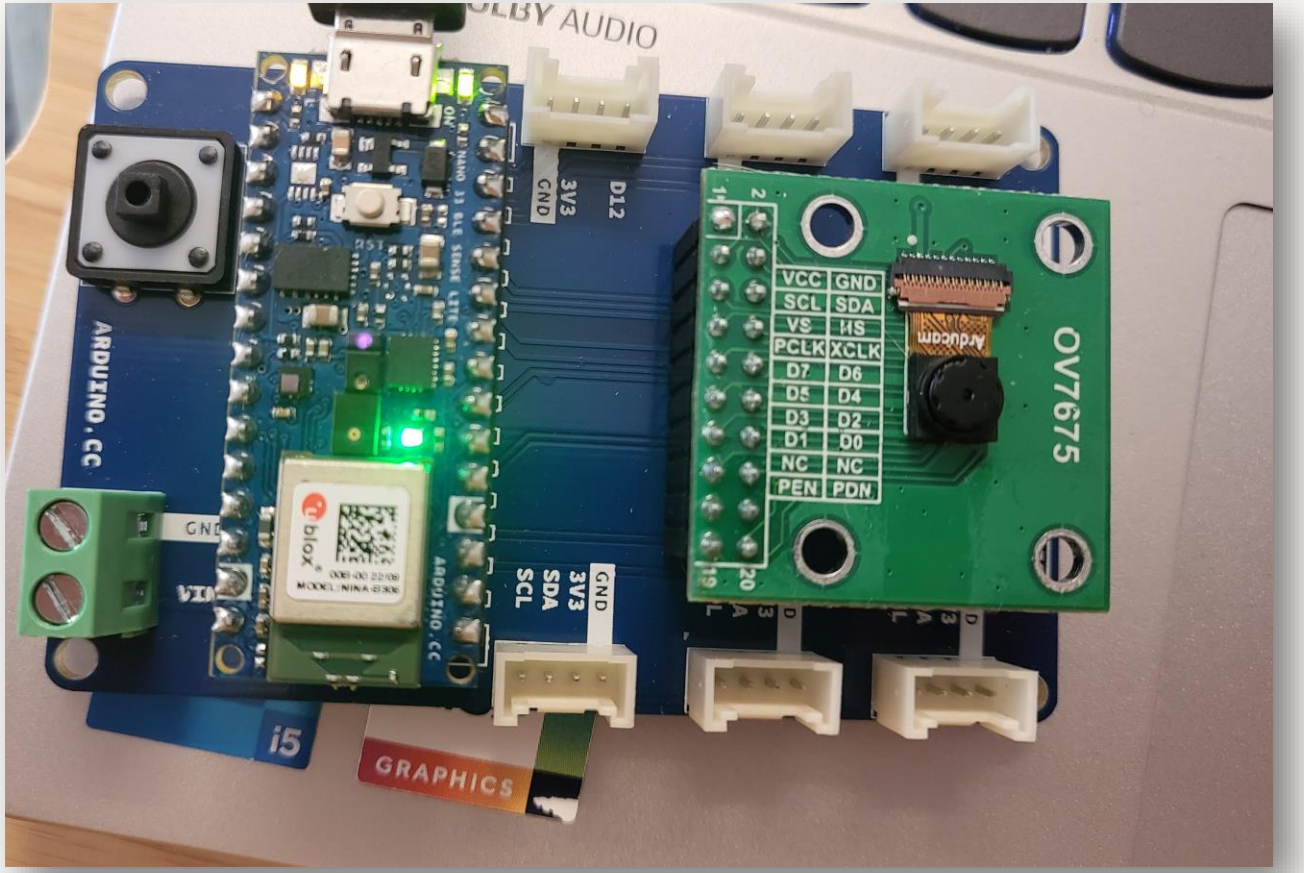


Mqtt klijent i BLE servis

BLE konekcija

Notifikacija za "detekciju" osobe





Arduino kao akturator

Slučaj kada su parking senzori aktivirani

Hvala na pažnji!

Teodora Kocić

Teodora.kocic@elfak.rs