

**Diplomski studij**

**Informacijska i komunikacijska tehnologija**

**Računarstvo**

Telekomunikacije i informatika  
Obradba informacija  
Računalno inženjerstvo

**Internet stvari  
  
EVID3NT – sustav za praćenje prisutstva na nastavi**

**Projekt**

**Karlo Dimjašević**

**Marin Pudić**

**Barbara Kralj**

**Marko Prosenjak**

**Fran Šribar**

**Lovro Gaćina**

**Ak.g. 2023./2024.**

**Sadržaj**

[1. Uvod 3](#_Toc41046714)

[2. Opis rješenja 3](#_Toc41046715)

[3. IoT platforma 4](#_Toc41046716)

[4. Korisničke aplikacije 4](#_Toc41046717)

# 1. Uvod

* Osnovno o usluzi
* Koji senzori?
* Koji aktuatori?
* Opisati min. 2 slučaja uporabe
  + *Primjer: upravljanje osvjetljenjem ovisno o prisutstvu korisnika u prostoriji i trenutnoj razini osvjetljenja; prikaz podataka o kvaliteti zraka u gradovima*
* Navesti slične projekte

# 2. Opis rješenja

* Opisati funkcionalnosti pojedinih komponenti
  + Senzorski čvor
  + (Aktuatori)
  + (Mrežni prilaz)
  + IoT platforma
  + Korisničke aplikacije
* Protokoli za komunikaciju između pojedinih komponenti
* *Primjer skice rješenja:*
  + *Navesti protokole koji se koriste*
  + *Navesti entitete u mreži (poslužitelj, mrežni prilaz, IoT uređaji sa senzorima, ...)*
  + *Prikazati vrste korisničkih aplikacija*
  + *Prilagoditi svojem rješenju, obavezno je imati poslužitelj (cloud platforma), korisničke aplikacije i senzore, mogu se koristiti različiti komunikacijski protokoli*
* Implementacijski detalji
  + Koji senzori/aktuatori su se koristili?
    - Tip senzora ako su se koristili fizički senzori
    - Opisati način simulacije ako su se senzorske vrijednosti simulirale
  + Kako je omogućena komunikacija između komponenti?
    - Koji radni okviri su se koristili (npr. *Californium* za CoAP, *Mosquitto* za MQTT i sl.)

Kao senzor je korišten RFID čitač RC522 spojen na mikrokontroler ESP32. Čitač služi za očitavanje privjesaka i kartica koji sadrže do 1 kB podataka. U ovom konkretnom slučaju, svaka kartica (u stvarnoj upotrebi bile bi korištene studentske iskaznice) ima ID koji se sastoji od 32 bita (u programu se svaki byte prevodi u odgovarajući dekadski broj, te se između svaka 2 broja nalazi crtica „-„). Čitač radi na frekvenciji od 15,36 MHz, a ima domet 10 cm.

Za izradu programskog rješenja korišten je Arduino IDE (korišten za pisanje programskog koda i kompajliranje koda), a sam program je napisan u programskom jeziku Arduino (*C/C++).* U programu se koriste deklaracije varijabli navedene u config.h datoteci (ne nalazi se u Github repozitoriju jer se u njoj nalaze konkretni podaci, poput lozinke za lokalnu mrežu, no predana je datoteka example\_config.h koja sadrži sve deklaracije, bez konkretnih vrijednosti – predan je template). Pri pokretanju programa, mikrokontroler se spaja na lokalnu Wi-Fi mrežu, koristeći neke od deklaracija navedenih u config.h datoteci. Nakon toga se postavlja MQTT server (uz pomoć informacija iz config.h) te se pokušava povezati na njega (MQTT\_CLIENT\_ID parametar je generiran na ThingsBoard te je pohranjen u config.h datoteku). MQTT server o kojem je riječ je *ThingsBoardov* MQTT server, koji koristi mosquitto kao radni okvir. Uređaj s ThingsBoard-om komunicira preko MQTT protokola. U petlji (loop) se na početku provjerava je li veza s Wi-Fiem ili s MQTT serverom prekinuta, te ako je, potrebno je provesti postupak povezivanja. Nakon toga, program čeka da senzor očita karticu ili privjesak (što se provjerava funkcijom rfid.PICC\_IsNewCardPresent() – u slučaju da nije otkrivena kartica, pozove se „return“, te se ne izvodi kod ispod te naredbe, nego petlja ponovno kreće ispočetka). U trenutku kada je jedno od toga skenirano, program provjerava je li ID prislonjene kartice (ili privjeska) jednak onome što je pohranjeno u nuidPICC (u nuidPICC se spremaju podaci o ID-u posljednje očitane kartice (ili privjeska)). Ovaj uvjet je potreban kako se ne bi zabilježilo skeniranje iste osobe više uzastopnih puta. U slučaju da se ne radi o istoj kartici (ili privjesku) koja je korištena prilikom prijašnjeg očitanja (dovoljno je da se ID razlikuje u jednom byte-u), rfid.uid.uidByte se pretvara u String, te se zajedno s učionicom u kojoj bi se uređaj trebao nalaziti, formatira kao JSON (primjer: {„cardId“:“10-147-255-231“, „classroom“:“b4“}). Dobiveni podaci se u obliku JSON-a šalju (publish-aju) na MQTT server na topic: v1/buildings/b/classrooms/b4*.* U projektu nije omogućeno povezivanje uz pomoć TLS-a, jer bi za ostvarivanje te mogućnosti bilo potrebno konfigurirati ThingsBoard, što nije moguće provesti bez pristupa računalu na kojem je pokrenut.