



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradbu informacija
Sveučilište u Zagrebu

Sustav za detekciju zauzetosti kupaonice



- Δ Za svakoga tko želi saznati nešto o Raspberry Pi 2 i detekciji zauzete prostorije
- Δ Za razumijevanje rada potrebne su osnove elektrotehnike i programiranja

Sažetak

S razvojem tehnologije imamo želju olakšati dijelove života za koje se prije smatralo da nije moguće ili čak da je nepotrebno. Nude nam se brojni načini olakšavanja života. Jedan od problema koji postoji je čekanje u redu na kupaonicu kako u poslovnim, tako i u privatnim prostorima. U ovom radu obrađuje se jedan od načina kako detektirati zauzetu prostoriju i način dojave korisnicima da je prostorija zauzeta. Sustav obrađen u ovom radu razvijen je na Raspberry Pi računalu i korišten je magnetski senzor za vrata. Od ovakvog projekta će osobito imati koristi uredi s malim brojem kupaonica za svoje radnike.

Sadržaj

1. UVOD	3
2. OPIS SUSTAVA I NJEGOVI DIJELOVI	4
2.1. Raspberry Pi 2	4
2.2. Magnetski senzor za vrata	5
3. REALIZACIJA SUSTAVA.....	6
3.1. Spajanje Raspberry Pi 2 i magnetskog senzora	6
3.2. Programska implementacija.....	7
4. ZAKLJUČAK.....	12
5. LITERATURA.....	13
6. POJMOVNIK	14

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Projekt pametne kuće je vrlo koristan i zanimljiv projekt kojim se olakšava svakodnevica i općenito život. S obzirom na to da se tehnologija iz dana u dan razvija i raste, daje nam mogućnost razvijanja pametnih rješenja kako bi postigli veću udobnost života. Omogućava nam zaštitu od provalnika, požara, olakšava brigu o ljubimcima i slično.

Osim navedenih prednosti, postoji i mnoge druge, za koje bismo možda rekli da nisu potrebni, ali jednom kada postoje ipak imamo korist od rješenja.

U takva rješenja spada ovaj projekt detekcije zauzetosti kupaonice. Svakodnevno na fakultetu i radnom mjestu se suočavamo s odlaskom do kupaonice i saznanja da je ona zauzeta. Vraćamo se na svoje mjesto te se nakon nekog vremena vraćamo provjeriti je li se prostor oslobodio.

Time se gubi mnogo dragocjenog vremena koje se moglo iskoristiti za druge aktivnosti. Ovim sustavom bi se to spriječilo.

Sustav izveden u ovom projektu se sastoji od Raspberry Pi 2 računala, magnetskog senzora za vrata i web aplikacije. Ideja sustava je na vrlo jednostavan način korisnicima na lokalnoj mreži nuditi informaciju o zauzetosti kupaonice tako da Raspberry Pi dohvaća uz pomoć senzora stanje vrata te se to stanje prenosi na web aplikaciju kojoj mogu pristupiti korisnici na toj lokalnoj mreži i vidjeti stanje.

Postavljanjem ovog sustava se rješava problem čekanja pred zauzetom kupaonicom. Ne mora se upotrebljavati samo za kupaonice, već i za neke druge prostorije koje se često koriste.

2. Opis sustava i njegovi dijelovi

Sustav se sastoji od Raspberry Pi 2 računala, magnetskog senzora za vrata i web aplikacije koja služi za praćenje procesa.

Postoje mnogi senzori za detekciju ljudske prisutnosti kao što su infracrveni, ultrazvučni, mikrovalni ili kapacitivni senzori. Odabran je magnetski senzor za vrata radi jednostavne nabave i implementacije.

2.1. Raspberry Pi 2

Raspberry Pi 2 je malo računalo s mnogim mogućnostima upotrebe. S njim se može pregledavati sadržaj na internetu, gledati video snimke visoke kvalitete, igrati igrice i mnoštvo drugih stvari.



Slika 1: Raspberry Pi 2

Raspberry Pi 2 ima 40 pinova za ulaz/izlaz koje odabire korisnik. Ima određene pinove za napajanje 3.3V i 5 V te pinove za uzemljenje.

Raspberry Pi2 GPIO Header			
Pin#	NAME		NAME Pin#
01	3.3v DC Power	●	DC Power 5v 02
03	GPIO12 (SDA1, PC)	●	DC Power 5v 04
05	GPIO13 (SCL1, PC)	●	Ground 08
07	GPIO14 (GPIO_GCLK)	●	(TXD0) GPIO14 08
09	Ground	●	(RXD0) GPIO15 10
11	GPIO17 (GPIO_GEN0)	●	(GPIO_GEN1) GPIO18 12
13	GPIO27 (GPIO_GEN2)	●	Ground 14
15	GPIO22 (GPIO_GEN3)	●	(GPIO_GEN4) GPIO23 16
17	3.3v DC Power	●	(GPIO_GEN5) GPIO24 18
19	GPIO10 (SPL_MOSI)	●	Ground 20
21	GPIO19 (SPL_MISO)	●	(GPIO_GEN6) GPIO25 22
23	GPIO11 (SPL_CLK)	●	(SPL_CEO_N) GPIO18 24
25	Ground	●	(SPL_CE1_N) GPIO17 26
27	ID_SD (PC ID EEPROM)	●	(PC ID EEPROM) ID_SC 28
29	GPIO15	●	Ground 30
31	GPIO16	●	GPIO12 32
33	GPIO13	●	Ground 34
35	GPIO19	●	GPIO16 38
37	GPIO26	●	GPIO20 38
39	Ground	●	GPIO21 40

Slika 2: Raspored pinova u Raspberry Pi 2

2.2. Magnetski senzor za vrata

Magnetski senzor za vrata se ponaša kao sklopka. Kada je zatvorena strujni krug je zatvoren i teče struja, a kada se sklopka otvori strujni krug je otvoren i struja ne teče.

Ovim senzorom se postiže detekcija otvorenosti vrata. Kada se senzor postavi u rub vrata zatvaranjem, odnosno otvaranjem vrata se zatvara, odnosno otvara strujni krug te to služi kao indikator zauzetosti prostorije.



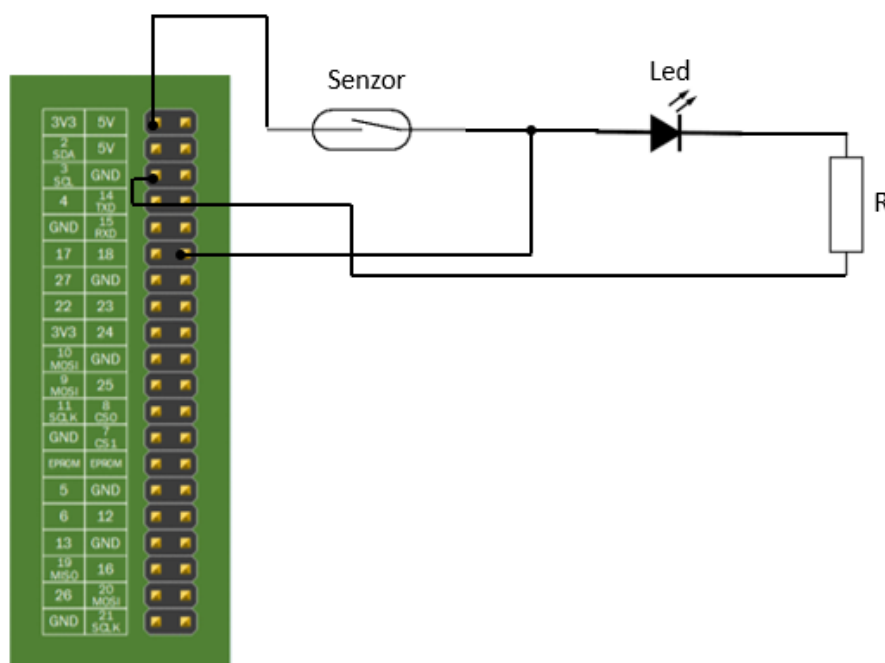
Slika 3: Magnetski senzor za vrata

3. Realizacija sustava

Za početak je potrebno spojiti komponente tako da LED lampica posluži kao indikator zatvorenog/otvorenog kruga. Nakon provjere ispravnog rada sustava slijedi povezivanje s računalom i izrada i povezivanje s web aplikacijom. Nakon provjere osnovnog rada potrebno je omogućiti praćenje zauzetosti kupaonice web aplikacijom. Treba omogućiti dodavanje korisnika u red čekanja i dobivanja obavijesti kada se kupaonica oslobodi.

3.1. Spajanje Raspberry Pi 2 i magnetskog senzora

Magnetski senzor za vrata treba biti spojen tako da prekida strujni krug kada su vrata otvorena, odnosno kada su magneti senzora fizički odvojeni. Shema spajanja kruga prikazana je na slici 4. Dodatno, u strujni krug je uključena LED lampica kao indikator ispravnosti rada.

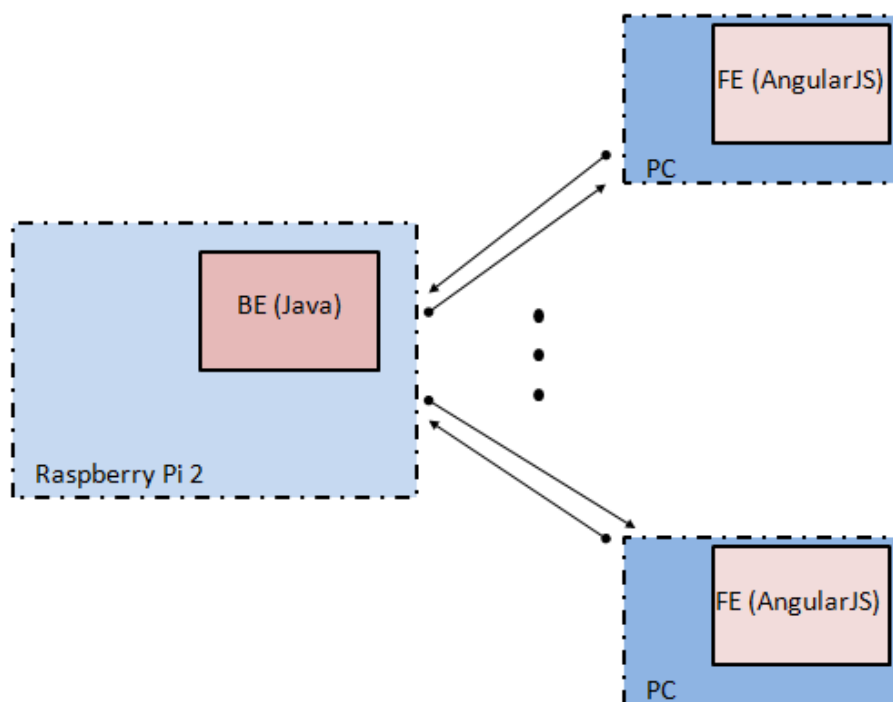


Slika 4: Električna shema

3.2. Programska implementacija

Sustav je koncipiran na način tako da Raspberry Pi 2 predstavlja server. Raspberry Pi očitava stanje pina 18 te ako je stanje *low* vrata su otvorena i kupaonica je slobodna, odnosno ako je stanje *high* kupaonica je zauzeta. Backend dio aplikacije pisan je u Java programskom jeziku. Za implementaciju frontend dijela aplikacije koristi se AngularJS. AngularJS je *JavaScript framework*. On služi za izradu aplikacijske logike i omogućava manipulaciju elemenata stranice.

Arhitektura sustava prikazana je na slici 5.



Slika 5: Arhitektura sustava

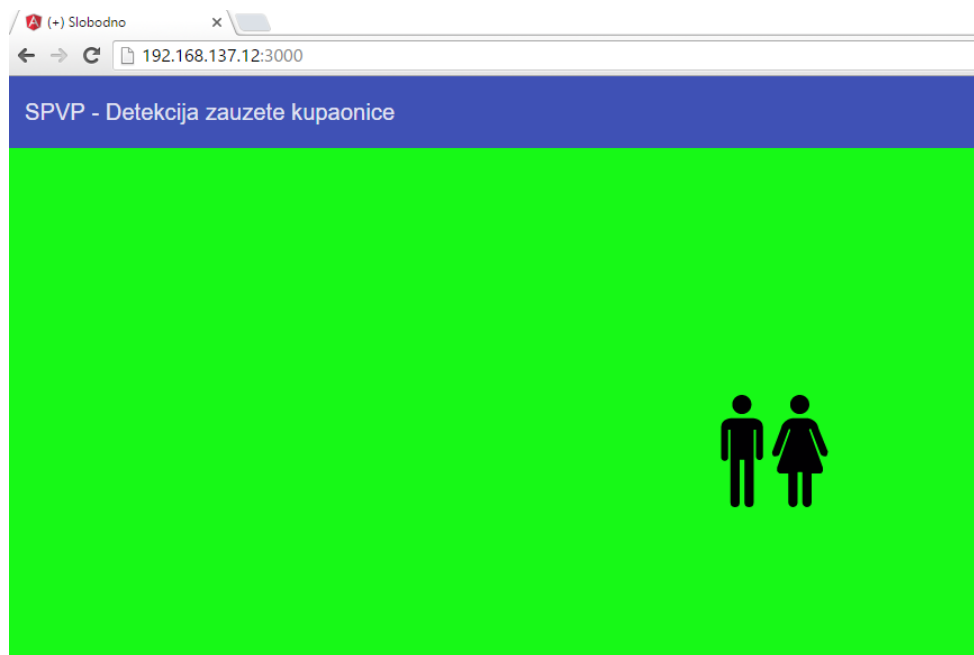
Definirano je nekoliko događaja koje je potrebno implementirati. Pri prvom učitavanju stranice se počinje s provjeravanjem stanja senzora. FE (frontend) šalje zahtjev na BE (backend) koji provjerava stanje senzora i vraća informaciju. Ta informacija se čuva u scope varijabli u angularu.

```
// Funkcija koja provjerava je li kupaonica slobodna
vm.isFree = () => {
  if (vm.waiting) {
    return;
  }

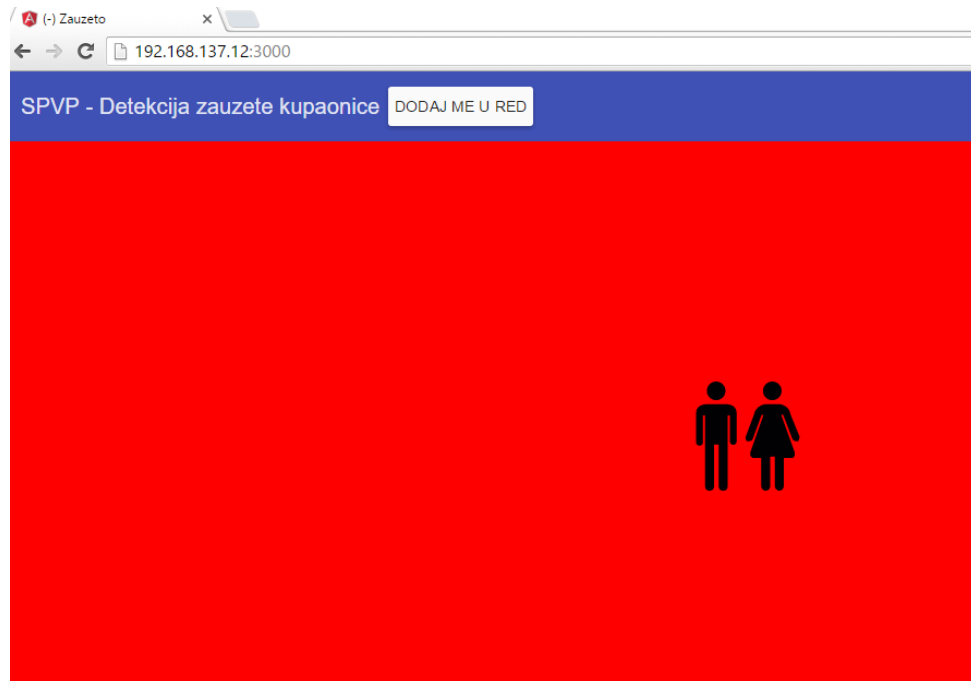
  HttpUtilities.get('/api/state')
    .then((data) => {
      vm.free = (data !== "high");

      // Ako je kupaonica slobodna ili je vec dodijeljena trenutnoj osobi
      // tada prikazi korisniku "(+) Slobodno"
      if (vm.free || vm.onDuty) {
        vm.notify = true;
      } else {
        // Inace prikazi "(-) Zauzeto"
        vm.notify = false;
      }
    });
};
```

Uz pomoć tog stanja razlikujemo zauzetu od nezauzete kupaonice. To se na ekranu manifestira kao crveni (slika 7), odnosno zeleni (slika 6) ekran. Kako bi se omogućilo dodavanje korisnika u red čekanja ako je kupaonica zauzeta, dodana je tipka na ekran koja se prikazuje samo kada je kupaonica zauzeta (u suprotnom je kupaonica slobodna te nije ni potrebno dodavanje u red).



Slika 6: Izgled web aplikacije kada je kupaonica slobodna



Slika 7: Izgled web aplikacije kada je kupaonica zauzeta

Klikom na tipku *Dodaj me u red* prvo se šalje HTTP-GET zahtjev na BE za dohvat ID korisnika.

```
// HTTP GET zahtjev za dohvat jedinstvenog IDja
HttpUtilities.get('/api/id')
  .then((data) => {
    // Spremi ID na scope
    vm.id = data;
  });
```

Taj ID je jedinstven za svakog korisnika i dodjeljuje se pritiskom na tipku. U dolje navedenom programskom odsječku prikazan je servis koji je zadužen za generiranje identifikacijskog broja.

```
/**
 * Servis zaduzen za sigurno generiranje ID-jeva.
 */
public class IdService {
    private static AtomicInteger counter = new AtomicInteger(0);

    // Privatni konstruktor
    private IdService() {
    }

    // Povecava vrijednost varijable te dohvaca novi id
    public static int getNextId() {
        return counter.incrementAndGet();
    }
}
```

U IdHandler komponenti se vrši poziv prema IdService komponenti čime se vrši dohvat novog identifikacijskog broja koji se prosljeđuje na FE. U ovom pozivu također se novostvoreni ID dodaje u red čekanja, pri čemu je se red bazira na FIFO pravilu (first-in-first-out).

```
/**
 * Vraca jedinstveni ID kao odgovor.
 */
public class IdHandler implements HttpHandler {

    @Override
    public void handle(HttpExchange httpExchange) throws IOException {
        // Zatrazi novi jedinstveni id od servisa
        int id = IdService.getNextId();

        // Pretvori id u string
        String response = String.valueOf(id);

        // Dodaj ID u red cekanja
        QueueService.registerId(id);

        // Posalji id na FE kao odgovor
        httpExchange.sendResponseHeaders(200, response.length());
        OutputStream os = httpExchange.getResponseBody();
        os.write(response.getBytes());
        os.close();
    }
}
```

Nakon što je jedinstveni ID dodijeljen korisniku (na FE), klijent repetitivno postavlja zahtjev za provjerom stanja reda. U tom zahtjevu se provjerava jesu li vrata otvorena i je li prosljeđeni ID sljedeći na redu. Ta informacija se na FE prosljeđuje kao „true“ ili „false“. Programski odsječak prikazan je u nastavku.

```
// Provjeri stanje vrata
GpioState gpioState = GpioState.getInstance();
boolean state = gpioState.getPinState();

// Dohvati glavu reda
int currentId = QueueService.getFront();

// Ako je dani id jednak glavi reda i vrata su otvorena
if (currentId == id && !state) {
    response = "true";
} else {
    response = "false";
}
```

Stanje reda se provjerava svake sekunde. Kada je trenutna osoba na redu, FE obavještava korisnika koji posjeduje ID sljedeći na redu o tome da je sada kupaonica slobodna i trenutno rezervirana za tog korisnika. Time dobivamo obavijest na web stranici, u tekstu prozora se pojavi (1), odnosno u nastavku piše *Slobodno*, dok u suprotnom slučaju dok je kupaonica zauzeta stoji (0) i u nastavku *Zauzeto*.

Red se neće pomaknuti na sljedeću osobu (odnosno kupaonica osloboditi) dok god se korisnik ne odjavi na aplikaciji bez obzira na otvorena/zatvorena vrata. Kada se korisnik odjavi šalje se zahtjev za brisanje njegovog identifikacijskog broja iz reda.

```
// Dohvati glavu reda
int currentId = QueueService.getFront();

// Ako je glava reda jednaka danom id-ju; pobrisi iz reda
if (currentId == id) {
    QueueService.removeFront();
}
```

Nakon toga se prelazi na sljedeći ID ukoliko postoji. Ako nema više nikoga u redu i ako su vrata otvorena onda je stanje kupaonice pri pristupu aplikaciji *Slobodno*.

4. Zaključak

Ovim sustavom se rješava nepotrebno čekanje na red za kupaonicu. Također omogućuje rezervaciju prostorije što se može iskoristiti za druge prostorije koje su često u upotrebi. Od ovog sustava najviše koristi bi imali poslovni prostori i uredi gdje se stvara gužva pri odlascima do kupaonica.

Sljedeći korak bio bi unaprjeđenje same detekcije zauzetosti prostorije, na primjer infracrvenim senzorom za detekciju ljudske prisutnosti budući da magnetski senzor za vrata zahtijeva zatvaranje/otvaranje vrata korisnika iza sebe što nije u svim situacijama precizno i moglo bi stvarati poteškoće.

Također, sustav se može proširiti na praćenje više prostorija istovremeno te dodavanjem dodatne obavijesti korisniku kada se kupaonica oslobodi, primjerice dolaskom obavijesti na mail ili dobivanjem sms poruke.

5. Literatura

- [1] Is the toilet free?,2014.
URL: <http://madebymany.com/blog/is-the-toilet-free>
- [2] The Pi4J Project, 2015.
URL: <http://pi4j.com>
- [3] Security Sensors, 2015.
URL:<http://rocode.com/sensors/>
- [4] Seed-angular-node, 2016.
URL:<https://github.com/mskec/seed-angular-node>

6. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Raspberry Pi 2	Malo računalo sa mnogim mogućnostima i primjenama	https://www.raspberrypi.org/
Magnetski senzor za vrata	Senzor koji radi kao sklopka, kada se magneti spoje zatvori se strujni krug.	https://en.wikipedia.org/wiki/Reed_switch
AngularJS	Javascript framework. Frontend development.	https://angularjs.org/