

Ivan Moštak  
003648500

Dolores Jurić  
0036484344

Petra Bucić  
0036484984  
Antonija Ćesić  
0036484302

# Smart Key

SEMINARSKI RAD - SPVP



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



- △ Namijenjeno svim zainteresiranim
- △ Povezivanje komponenti
- △ Programiranje funkcionalnosti pomoću Arduino mikrokontrolera
- △ Opisi i upute za izradu modula sustava

## Sažetak

„Smart Key“ je sustav za upravljanje vratima. Njegova svrha je zaštita prostorija korištenjem nekih od podsustava za autorizacije korisnika. U isto vrijeme sustav radi i kao asistent u kućanstvu te izvršava upravljačke sekvence za otključavanje i zaključavanje te otvaranje i zatvaranje vrata. Kod tradicionalne izvedbe vrata svaki korisnik mora imati ključ koji mu se fizički mora predati u vrijeme kada ima dozvolu korištenja. Temeljna ideja ovog sustava je omogućiti udaljeno upravljanje pristupom te korisnici više ne moraju imati fizički ključ već im je omogućen pristup preko RFID kartice, otiska prsta ili mobilne aplikacije, ovisno o razini prava koju imaju. Na taj način štedi se vrijeme i povećava transparentnost korištenja što je u interesu vlasniku prostora, ali svima ostalima koji koriste sustav, a nemaju skrivene namjere. Nedostatak ovog sustava je ovisnost o napajanju te potreban spoj na mrežu koji ga može učiniti ranjivim na napade.

## Sadržaj

1. UVOD .....	3
2. „SMART KEY“ SUSTAV .....	4
2.1. RFID (Ćesić) .....	5
2.2. Otisak prsta (Jurič).....	5
2.3. Baza korisnika (Moštak) .....	6
2.4. Mobilna aplikacija (Bucić) .....	7
2.5. Arduino.....	8
2.6. Izvršni uređaji (Bucić).....	8
3. SUSTAV ZA AUTORIZACIJU I UPRAVLJANJE VRATIMA.....	9
3.1. Prepoznavanje korisnika preko RFID-a (Ćesić) .....	9
3.2. Prepoznavanje korisnika preko otiska prsta (Jurič) .....	13
3.3. Prepoznavanje korisnika preko web servisa (Moštak) .....	19
3.4. Provjera korisnika pomoću SD kartice (Moštak) .....	22
3.5. Zadavanje uputa preko mobilne aplikacije (Bucić) .....	24
3.6. Otključavanje i zaključavanje (Bucić) .....	27
3.7. Otvaranje i zatvaranje (Bucić) .....	28
4. ZAKLJUČAK.....	33
5. LITERATURA.....	34
POJMOVNIK .....	35

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

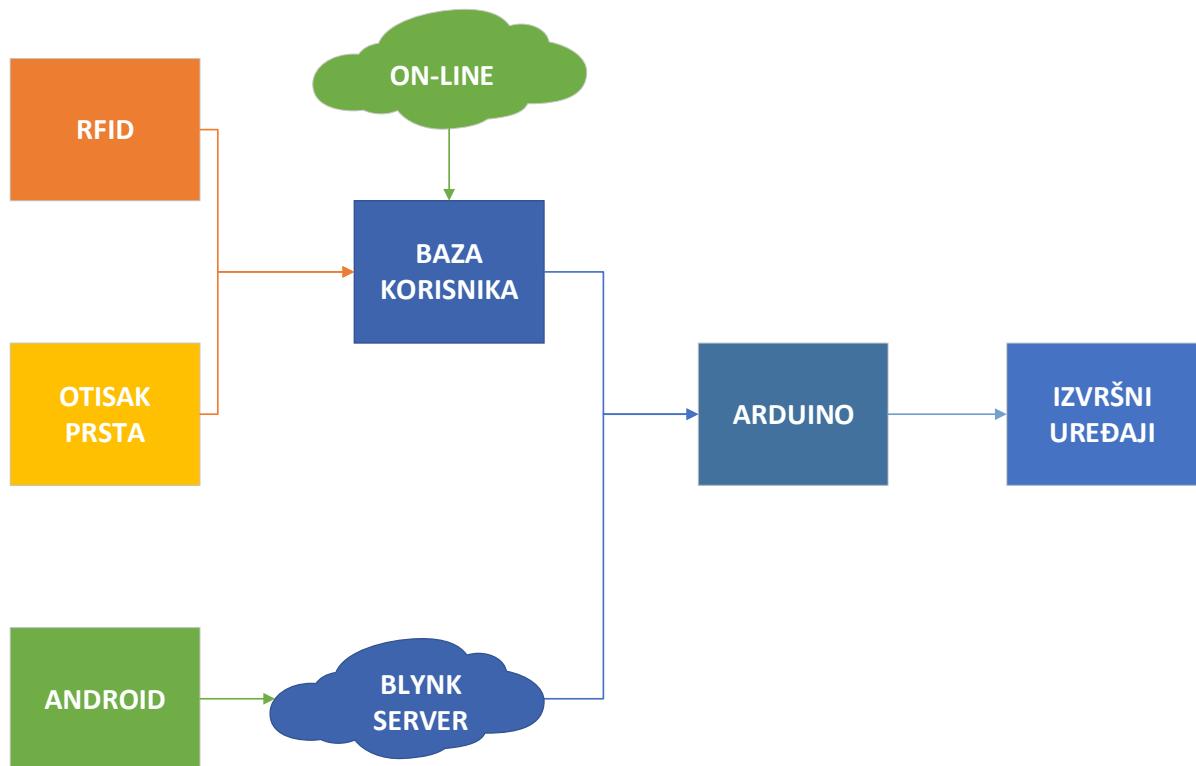
Razvojem mikrokontrolera otvaraju se razne mogućnosti implementacije novih funkcija u uređaje koji su se dugi niz godina koristili na tradicionalne načine. Tako se uz malo mašte i najobičniji uređaji kao što su vrata mogu pretvoriti u „pametne“. To je važno svim korisnicima prostora, a posebno kada istim prostorijama ima pristup više ljudi i kada se pravo pristupa mijenja. Stvaranjem dinamičke baze korisnika može se točno kontrolirati kada netko smije ući u prostor, za razliku od situacije kada se nekome da ključ te se više nema nadzor nad tim kako ga on koristi. Sustav također rješava problem zaboravljanja ključeve te korisniku štedi vrijeme. Uz sigurnost, sustav za otključavanje i otvaranje vrata olakšava korištenje. To je bitno kada je korisniku otežan pristup, na primjer zbog stvari koje nosi ili u slučaju osoba s poteškoćama u kretanju. Udaljeno otvaranje vrata omogućuje da korisnik prođe kroz vrata bez zaustavljanja i problema oko otključavanja i otvaranja vrata.

U radu se proučavaju načini na koje se može povećati sigurnost prostora, upravljati pravom pristupa te olakšati korištenje. U prvom poglavlju dan je pregled komponenti „Smart Key“ sustava te njihova povezanost, dok je u drugom poglavlju objašnjena implementacija i funkcionalnost svakog podsustava. Korištene komponente su RFID, čitač otiska prsta, Web server temeljen na Apache i MySQL alatu, mobilna aplikacija korištenjem Blynk platforme te mikrokontroler Arduino Uno koji na temelju njih šalje upute izvršnim uređajima, odnosno servo i DC motoru.

## 2. „Smart Key“ sustav

„Smart Key“ sustav sastoji se od podsustava za autorizaciju te upravljanje vratima. Prije nego se korisniku dozvoli upravljanje bravom i vratima potrebna je njegova autorizacija. Ovisno o tipu korisnika moguće je nekoliko načina potvrde identiteta. Cilj ovog projekta je bio demonstrirati moguće koncepte i njegovu funkcionalnost. Tako su implementirani pristup preko mobilne aplikacije, otiska prsta i RFID čitača. Uz to predstavljeni su načini izgradnje baze korisnika.

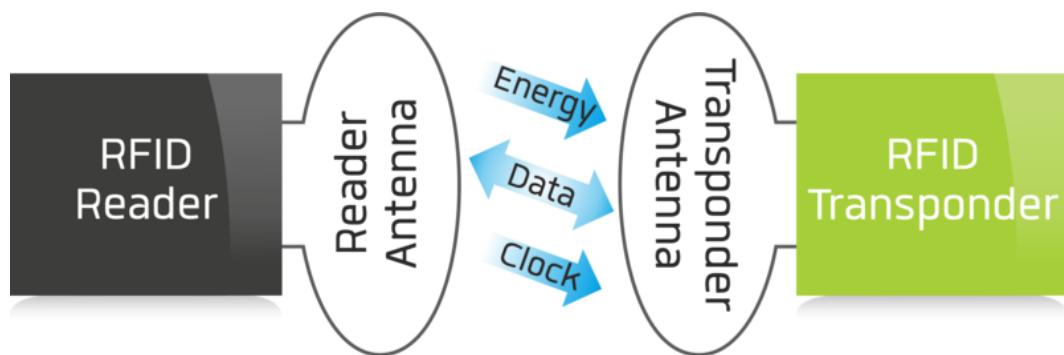
Sustav je zamišljen modularno, odnosno tako da korisnik može izabrati koje module želi uključiti i koliku razinu sigurnosti će imati na pojedinim vratima. Shematski prikaz povezivanje modula prikazan je na Slika 1.



**Slika 1: Struktura povezivanja modula sustava**

## 2.1. RFID (Ćesić)

Kada je u pitanju otključavanje vrata poželjna je visoka razina sigurnosti no ponekad je poželjno i da se ta radnja odvija brzo pogotovo ako se kroz ta vrata prolazi često. To se postiže RFID tehnologijom gdje je ključ u obliku kartice. Radio frekvencijska identifikacija (RFID) je metoda automatske identifikacije, koja je u posljednjih nekoliko godina pronašla široku primjenu na raznim područjima. Ovdje se radi o beskontaktnoj komunikacijskoj tehnici koja prenosi informacije za identifikaciju osoba, životinja, dobara i roba. Korištenje RFID-a omogućuje nam razne primjene koje s dosadašnjim identifikacijskim sistemima nismo mogli ostvariti. Jedan RFID sustav sastoji se s jedne strane od jednog nosača podataka (zvan Transponder ili TAG) i s druge strane od jednog pisača/čitača s antenom. RFID radi s slabim elektromagnetskim valovima koji se očitavaju pomoću čitača. Ukoliko se transponder donese u područje dohvata antene onda se mogu očitati ili pohraniti podaci s transpondera bez ikakvog dodira. Postoje dvije vrste transpondera: pasivne (nemaju vlastito napajanje već crpe svoju energiju izravno iz energetskog polja čitača i aktivne (puno više složeniji u odnosu na pasivne te imaju svoje vlastito napajanje (bateriju ili akomulator) koja omogućuje puno veći doseg čitanja). Na slici je prikazana pojednostavljena shema korištenog sustava.



Slika 2: RFID sustav

## 2.2. Otisak prsta (Jurić)

Otisak prsta svake osobe je jedinstven te se ne može zaboraviti, izgubiti ili pogoditi poput obične lozinke. Običan ključ za vrata je također jedinstven, ali se može izgubiti ili neovlašteno multiplicirati. Stoga identifikacija osobe pomoću otiska prsta predstavlja dodatnu razinu zaštite

od neovlaštenog pristupa nekoj prostoriji. Drugi razlog korištenja ove metode zaštite je ušteda vremena, budući da je brže i praktičnije prisloniti prst umjesto upisivanja lozinke.

Općenito, ovakav sustav za identifikaciju se sastoji od senzora za čitanje otiska i DSP procesora koji pohranjuje otiske u bazu podataka te na kojem se implementiraju algoritmi za prepoznavanje i analizu elemenata definiranih u bazi podataka.

Korišteni senzor svrstava se u optičke. Postavljanjem prsta na staklenu površinu, osvjetjava se površina prsta te se zatim mjeri reflektirana svjetlost i stvara digitalni otisak. Prilikom pretraživanja odgovarajućeg otiska u bazi podataka, analiziraju se preklapanja najsvjetlijih i najtamnijih područja. Problem ove tehnologije je što se što se pohranjuje samo 2D slika te što nečistoće ili ogrebotine na dodirnoj površni senzora smanjuju točnost prepoznavanja otiska.

### **2.3. Baza korisnika (Moštak)**

Baza korisnika implementirana je na dva načina. Online korištenjem Apache web servera i MySQL baze podataka i na arduinu, preko spojenog modula, micro SD kartice, na kojem se također nalazi „offline“ baza podataka. Ovako je sustav osiguran od prekida veze arduina s okolnim mrežama i omogućuje rad bez vanjske komunikacije.

Online baza podataka zamišljena je i kao upravljačka jedinica za korisnike, tj. nude se mogućnosti kontrole, modifikacije kompletne baze podataka (slika 3). Moguće je dodati nove korisnike i modificirati tip njihove dozvole. Moguće je za svakog korisnika odrediti u kojem vremenskom intervalu ima pravo na otključavanje vrata (npr. puštanje spremićice u određeno doba radnog dana, 10:00-14:00 ili dozvola ulaska za radnike u točno određenom intervalu 07:00-08:15).

The screenshot shows a web application interface for managing user access permissions. At the top, there are two buttons: "Stvor novog korisnika" (Create new user) on the left and "Logout" on the right. Below these buttons, the title "Korisnička baza" (User base) is displayed. A table lists 10 users, each with an ID, permission level, and specific access times. At the bottom of the table is a blue "DELETE" button.

Id	Dozvola	Vrijeme ulaz (od)	Vrijeme ulaz (do)
1	1	12:00:00	14:00:00
2	1	00:00:00	23:59:59
3	0	00:00:00	00:00:00
4	1	12:00:00	14:00:00
5	1	08:00:00	14:00:00
6	0	00:00:00	00:00:00
7	1	10:00:00	16:00:00
8	1	00:00:00	23:59:59
9	1	00:00:00	23:59:59
10	1	08:00:00	16:00:00

**DELETE**

**Slika 3. baza korisnika preko web servisa**

Prednost ovakvog sustava koji ima dvije komponente je svakako u mogućnosti da se vlasnik ovakvog sustava (s pristupom internetu) u bilo kojem trenutku može ulogirati u upravljačku jedinicu i modificirati bazu korisnika, bez da se nalazi u blizini samog sustava. Prednost postojanja „offline“ baze korisnika je svakako u brzini izvođenja pretrage same baze nakon što se zatraži provjera korisnika od autorizacijskih elemenata RFID-a, čitača otiska prstiju ili dr. U slučaju da zbog nekog nepredviđenog razloga dođe do prekida u komunikaciji sustava s internetskom mrežom, sustav će i dalje raditi kako treba, jedino neće biti omogućena modifikacija baze korisnika.

Na web serveru je baza implementirana u besplatnom open-source MySQL sustavu za upravljanje podataka. MySQL baza ima podršku mnogih programskih jezika pa tako i za PHP, kojim je pisan programski kod web servisa za upravljačku jedinicu korisničke baze podataka.

Na arduinu je baza podataka ostvarena korištenjem „Extended database library“ čime se povećava maksimalni broj zapisa sa 256 na teoretski maksimum od 4,294,967,295 zapisa. SD kartica komunicira s Arduinom putem SPI komunikacije.

Obje baze podataka su zaštićene od potencijalnih zlonamjernih činitelja. Za pristup web upravljačkoj jedinici potrebno je znati odgovarajuće korisničko ime i lozinku, a pristup „offline“ bazi podataka uopće nije moguć osim preko web-a (u slučaju da korisnik ne želi koristiti web upravljačku jedinicu, postoji programski odsječak za arduino kojim se može direktno upravljati korisničkom bazom podataka.)

Komunikacija između web servera i arduina trenutno nije ostvarena te se stavlja kao prioritet za sljedeći korak u integraciji cijelog sustava u cjelinu. Nakoj integracije i povezivanja arduina sa web serverom (preko ethernet shielda) baze podataka korisnika će se redovito moći sinkronizirati.

## 2.4. Mobilna aplikacija (Bucić)

Mobilna aplikacija je jedan od mogućih načina registracije korisnika. U hijerarhiji sustava ovo se može smatrati podsustavom s najnižim stupnjem zaštite. Nekoliko je razloga za to, a prvi od njih je sama izvedba upravljanja. U ovom slučaju daje se najveća sloboda korisniku koji može samostalno upravljati ključanicom i vratima uz zadovoljene uvjete pozicije. Drugi je razlog to što se na ovaj način ne mogu razlikovat korisnici pa ne postoji mogućnost nadogradnje sustava kojom bi se pratile aktivnosti korisnika. Posljednji razlog je taj što se za izradu upravljačkog sučelja, odnosno komunikaciju između Androida i Arduina koristi Blynk platforma [2]. Budući da komunikacija ide preko servera nad kojim

nemamo nikakav nadzor korisniku se savjetuje da ovu mogućnost koristi za prostore koji nisu pod sigurnosnim rizikom.

## 2.5. Arduino

Svi moduli spajaju se preko Arduino mikrokontrolera te on predstavlja središnji čvor koji služi kao posrednik između upravljačkih i izvršnih uređaja. Funkcionalnost cijelog sustava uvelike je uvjetovana kodom koji se nalazi na mikrokontroleru. Tako se u kodu definiraju načini autorizacije te sekvenca koja se izvršava nakon toga. Također, potrebno je podesiti komunikaciju tako da se mogu primati podatci sa servera. U stvarnoj implementaciji bilo bi potrebno serijsku komunikaciju preko računala, koja se koristi u primjeru, zamijeniti direktnim spajanjem na mrežu.

## 2.6. Izvršni uređaji (Bucić)

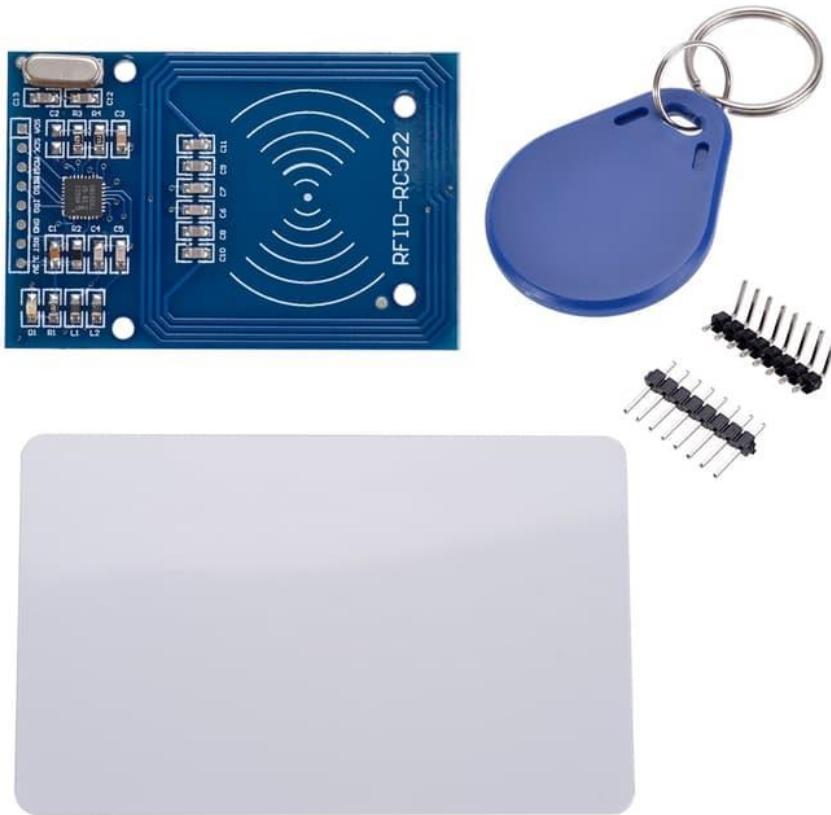
Tip i snaga izvršnih uređaja ovise o tome kakva su vrata na koja se instalira uređaj. Više o mogućim izvedbama brava može se pronaći na [7]. U ovom projektu će se prepostaviti da se brava može otključati zakretanjem cilindra. U tom slučaju moguće je koristiti servo motor kojeg bi se potom ugradilo oko cilindra tako da se ostvari njegovo zakretanje zajedno s vratilom motora. Odabir aktuatora za otvaranje vrata još je više ovisan tipu vrata pa će tako snaga motora ovisiti o masi i veličini vrata. Uz to, treba uzeti u obzir i način otvaranja. U općem slučaju otvaranje vrata može se postići upotrebom linearnog aktuatora ili rotacijskog DC motora uz dodane prijenosnike gibanja [6]. U ovom se projektu pokazuje koncept upravljanja te se stvarna ugradnja neće detaljnije razmatrati.

### 3. Sustav za autorizaciju i upravljanje vratima

U ovom poglavlju objašnjena je fizička i programska implementacija opisanih modula iz prethodnog poglavlja. Moduli zajedno čine Smart Key sustav koji omogućava autorizaciju korisnika te upravljanje bravom i samim vratima. Korisnik se može prijaviti putem RFID-a, otiska prsta ili mobitela. Kada je korisnik prepoznat izvršit će se neka od upravljačkih sekvenci ili će se samom korisniku prepustiti da upravlja vratima.

#### 3.1. Prepoznavanje korisnika preko RFID-a (Česić)

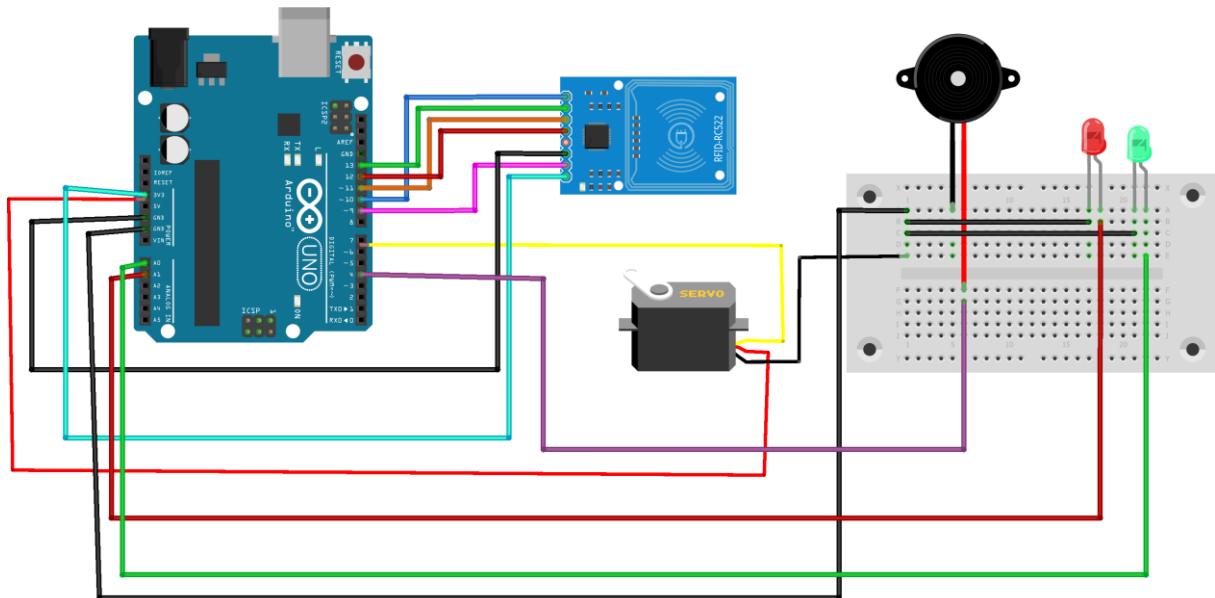
U projektu je korišten RFID modul RC522 koji se sastoji od komponenata: RFID MF-RC522 čitač/pisač, RC522 prazna bijela kartica, RC522 ključ i header 1x8. Komponente su prikazane na slici 4.



Slika 4: RFID komponente

Mikrokontroler i čitač kartica koriste SPI protokol za komunikaciju. Čitač kartica i oznake komuniciraju koristeći 13,56 MHz elektromagnetsko polje. Napaja se sa 3.3V s arduina i može očitati karticu s udaljenosti od 0 do

60mm. Spajanje RC522 modula s arduinom uno je prikazano na shemi prikazanoj na slici 5.



fritzing

**Slika 5: RFID sustav spojen na arduino**

Nakon što je provjereno da su sve komponente sustava dobro spojene napisana je funkcija koja u beskonačnoj petlji očitava pojavljuje li se i jedna kartica u blizini čitača. Nakon što se očita prisutnost kartice čeka se da čitač uspješno očita ID kartice. Nakon što su obje zastavice podignute provjerava se ID korisnika.

Postoji statička baza podataka koja sadrži ID-ove RC522 bijelih kartica ili ID RC522 ključeva svih korisnika. Ako je ID prislanjene kartice u sustavu i prolaz mu je dopušten upali se zeleno LED svjetlo, buzzer zazvoni zvonom za otvaranje i vrata se otključavaju pomoću servo mortora koji se napaja sa 5V s arduina. Ukoliko očitana kartica je u bazi ali joj je trenutno zabranjen ulaz pali se crveno Led svjetlo buzzer zazvoni zvonom za zabranu i vrata se ne otključavaju. Također se ispisuje poruka na serijskom sučelju da je pristup odbijen. Sustav ne reagira ako ID očitane kartice nije u bazi podataka. Kod je prikazan u nastavku

```
#include <deprecated.h>
#include <MFRC522.h>
#include <MFRC522Debug.h>
#include <MFRC522Extended.h>
#include <MFRC522Hack.h>
#include <require_cpp11.h>

#include "SPI.h"
#include "MFRC522.h"
#include <Servo.h>

#define SS_PIN 10
```

```
#define RST_PIN 9
#define LED_PINZ A0
#define LED_PINC A1
#define BUZZER 4
#define LockMotor 6
#define doorOpenTime 3000

Servo myservo; // create servo object to control a servo

int lock=1; //if locked wanted 1, if unlocked wanted 0
int pos = 0; // variable to store the servo position
long int t1=0;
long int t2=0;

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

void getRFID() {

    if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
        return;

    MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
    // Serial.println(rfid.PICC_GetTypeName(piccType));

    // Check is the PICC of Classic MIFARE type
    if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
        piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
        piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
        Serial.println(F("Your tag is not of type MIFARE Classic."));
        return;
    }

    String strID = "";
    for (byte i = 0; i < 4; i++) {
        strID +=
            (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
            String(rfid.uid.uidByte[i], HEX) +
            (i!=3 ? ":" : "");
    }
    strID.toUpperCase();

    Serial.print("Tap card key: ");
    Serial.println(strID);

    if (strID.indexOf("00:3E:81:A6") >= 0) { //card 00:3E:81:A6
        digitalWrite(LED_PINZ, HIGH);
        tone(BUZZER, 500);
        doorUnlock();
        Serial.print("ODOBRENO\n");
        delay(2000);
        digitalWrite(LED_PINZ, LOW);
        noTone(BUZZER);
        doorLock();
    } else if (strID.indexOf("82:94:D9:D9") >= 0) { //tag 82:94:D9:D9
        digitalWrite(LED_PINC, HIGH);
        Serial.print("ODBIJENO\n");
        tone(BUZZER, 100);
        delay(2000);
        digitalWrite(LED_PINC, LOW);
        noTone(BUZZER);
    }
}
```

```

    } else {
        digitalWrite(LED_PINZ, LOW);
        digitalWrite(LED_PINC, LOW);
    }

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
}

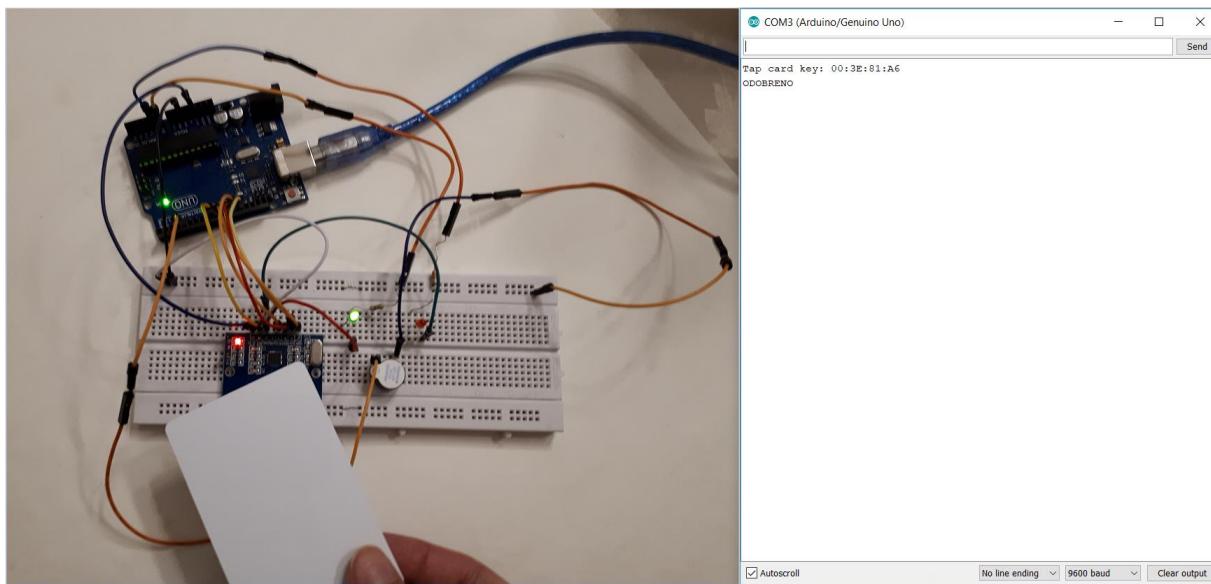
void setup() {
    Serial.begin(9600);
    SPI.begin();
    rfid.PCD_Init();
    pinMode(LED_PINZ, OUTPUT);
    pinMode(LED_PINC, OUTPUT);
    noTone(BUZZER);
    myservo.attach(LockMotor); // attaches the servo on pin 9 to the servo
object
    myservo.write(180);
    lock=0;
}

void loop() {

    getRFID();
    delay(50);
}

```

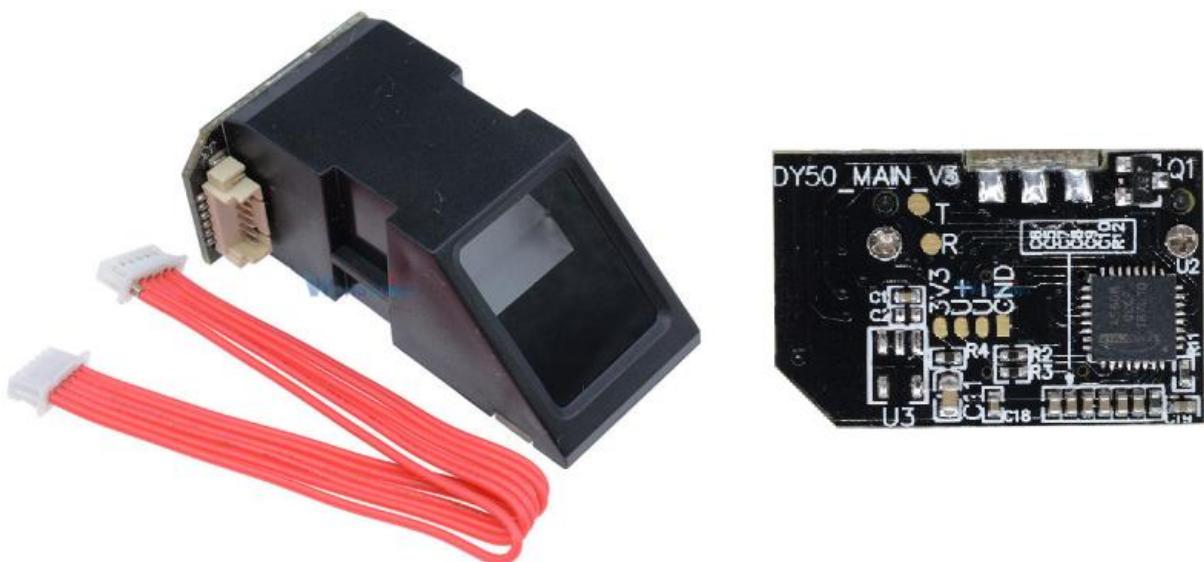
U funkciji je korištena biblioteka rfid-master. Također su korištene funkcije za zaključavanje i otključavanje vrata `doorLock()` i `doorUnlock()` obje funkcije su objašnjene kasnije u tekstu. Na slici 6. je prikazan primjer odobrenog prolaza.



**Slika 6: Prolaz odobren**

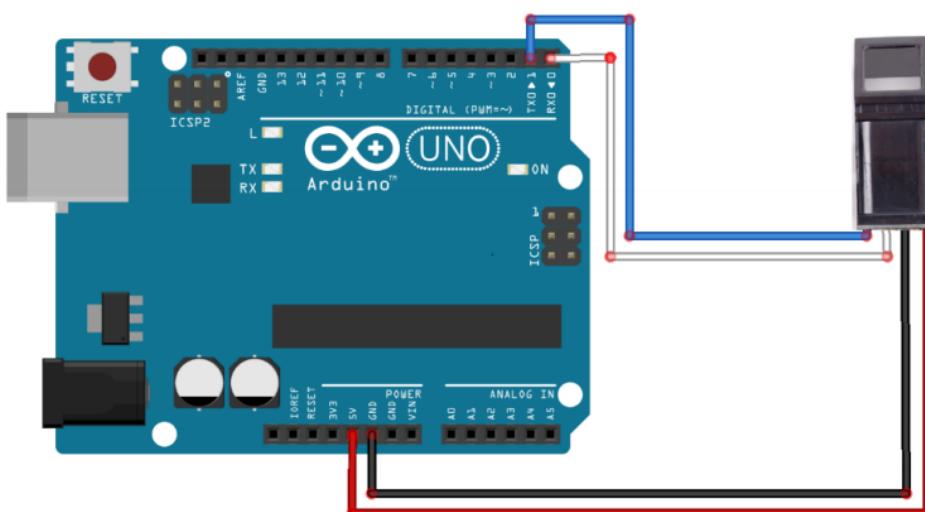
### 3.2. Prepoznavanje korisnika preko otiska prsta (Jurič)

Senzor koji se u projektu koristi za prepoznavanje korisnika preko otiska prsta je *Adafruit Fingerprint Sensor DY50*. Senzor je prikazan na slici 7 skupa s tiskanom pločicom na kojoj je DSP, koja se nalazi na njegovoj poleđini. Konektor koji je došao skupa sa senzorom nije bio kompatibilan s Arduinom, pa ga se s te strane moralo ukloniti i žice zalemiti na odgovarajući konektor (muško-ženski) na kojeg su se potom spojile žice različitih boja prilagođene svojoj funkciji. Također, na senzor je spojeno 6 crvenih žica, no 2 izvoda na tiskanoj pločici nemaju nikakvu funkciju, pa su se te 2 žice radi praktičnosti također uklonile.



**Slika 7: Senzor za prepoznavanje otiska prstiju**

Slika 8. prikazuje spajanje senzora s Arduinom. Crvena i crna žica su napajanje (+5V i masa), plava žica je TX, odnosno izlaz iz Arduina, a bijela žica je RX, odnosno ulaz sa senzora. Dakle, podaci se šalju serijskom komunikacijom Arduina (*SoftwareSerial*).



**Slika 8: Povezivanja senzora za otisak prsta s Arduinom**

Očitani otisci se spremaju u Flash memoriju čipa na senzoru. Moguće je pohraniti ukupno 127 otisaka. Ako se nekoj osobi želi dodijeliti autorizacija za otvaranje ili otključavanje vrata, potrebno je prvo unijeti

otisak prsta te osobe u bazu podataka na sljedeći način: nakon otvaranja serijskog prozora, od korisnika se traži da upiše broj koji predstavlja ID pod kojim će njegov otisak prsta biti spremišten. Korisnik zatim treba prisloniti prst na senzor, nakon čega senzor uzima sliku, potom odmaknuti prst te ponovno prisloniti prst. Ako su oba puta slike jednake, otisak prsta bit će spremišten pod definiranim ID-em. Navedena procedura provodi se pomoću Arduino programske datoteke 'register', koja je prikazana u nastavku. U kodu se koriste određene funkcije iz Adafruit Fingerprint biblioteke.

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // pin #2 is IN from sensor (BLUE wire)
                                // pin #3 is OUT from arduino (GREEN wire)
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup() {
    Serial.begin(9600);
    while (!Serial);
    delay(100);
    Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

    finger.begin(57600);

    if (finger.verifyPassword()) {
        Serial.println("Found fingerprint sensor!");
    } else {
        Serial.println("Did not find fingerprint sensor :(");
        while (1) { delay(1); }
    }
}

uint8_t readnumber(void) {
    uint8_t num = 0;

    while (num == 0) {
        while (! Serial.available());
        num = Serial.parseInt();
    }
    return num;
}

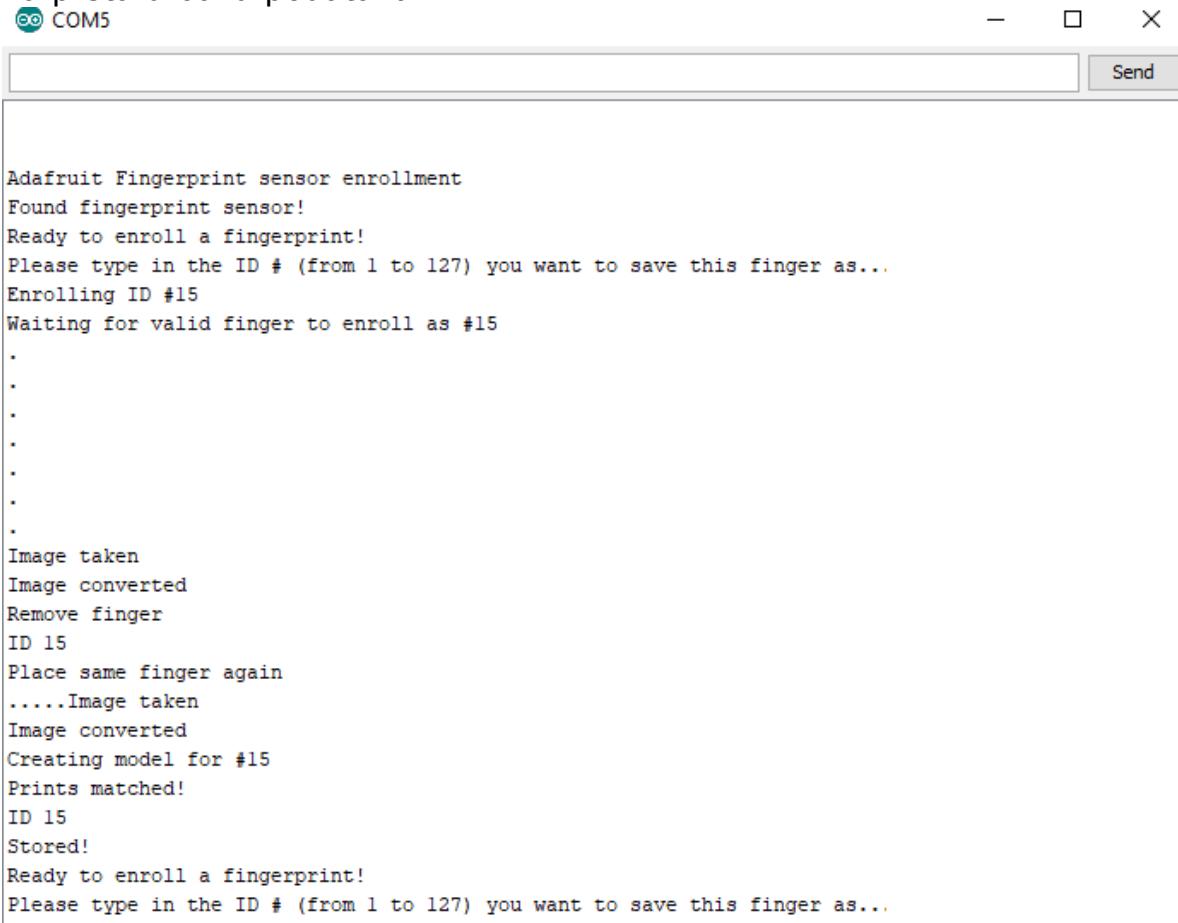
void loop() {
    Serial.println("Ready to enroll a fingerprint!");
    Serial.println("Please type in the ID # (from 1 to 127) you want to save this
finger as...");

    id = readnumber();
    if (id == 0) { // ID #0 not allowed, try again!
        return;
    }
    Serial.print("Enrolling ID #");
    Serial.println(id);

    while (! getFingerprintEnroll() );
}
```

```
uint8_t getFingerprintEnroll() {  
    int p = -1;  
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);  
    while (p != FINGERPRINT_OK) {  
        p = finger.getImage();  
        switch (p) {  
            case FINGERPRINT_OK:  
                Serial.println("Image taken");  
                break;  
            (...)  
        }  
    }  
  
    // OK success!  
    p = finger.image2Tz(1);  
    switch (p) {  
        case FINGERPRINT_OK:  
            Serial.println("Image converted");  
            break;  
        (...)  
    } (...)
```

Budući da je funkcija `getFingerprintEnroll()` poprilično dugačka, umjesto prikaza određenog dijela njezina koda naveden je znak (...). Slika 9 prikazuje poruke u serijskom prozoru prilikom procesa uvođenja novog otiska prsta u bazu podataka.

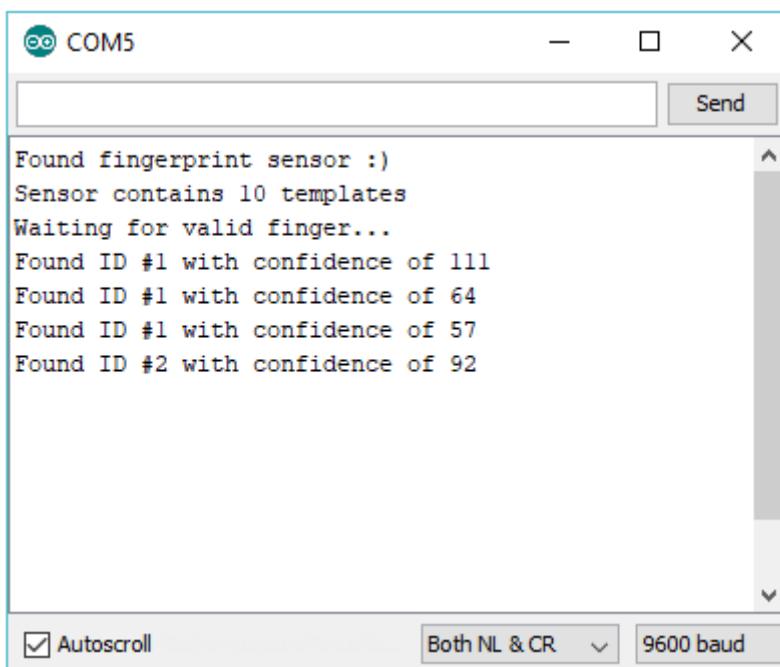


The screenshot shows a serial monitor window titled "COM5". The window has a header with a close button (X) and a send button. The main area displays the following text:

```
Adafruit Fingerprint sensor enrollment  
Found fingerprint sensor!  
Ready to enroll a fingerprint!  
Please type in the ID # (from 1 to 127) you want to save this finger as...  
Enrolling ID #15  
Waiting for valid finger to enroll as #15  
. . . . .  
Image taken  
Image converted  
Remove finger  
ID 15  
Place same finger again  
.....Image taken  
Image converted  
Creating model for #15  
Prints matched!  
ID 15  
Stored!  
Ready to enroll a fingerprint!  
Please type in the ID # (from 1 to 127) you want to save this finger as...
```

### Slika 9: Serijski prozor prilikom uvođenja novog otiska prsta

Određeni korisnik se prepoznaće pomoću 'fingerprint' programske datoteke (u nastavku), koja prepoznaće nalazi li se pojedini otisak prsta u bazi podataka i ovisno o tome ispisuje pripadajući ID i pouzdanost prepoznavanja otiska. Pouzdanost prepoznavanja (u kodu *confidence*) je mjera koliko se dobro očitani otisak prsta podudara s otiskom u bazi, a poprima vrijednosti 0-255. Slika 10. prikazuje zaslon serijskog prozora nakon što se prst prisloni na senzor. Poruka 'Found fingerprint sensor' označava da je krug sa slike 2 dobro spojen. U prozoru se također ispisuje poruka s informacijom koliko se trenutno otisaka prstiju nalazi u bazi.



### Slika 10: Serijski prozor prilikom prepoznavanja otiska prsta

Ako se korisnikov otisak nalazi u bazi podataka, pali se zeleno svjetlo na diodi i oglašava se zvuk koji predstavljaju zvučni i svjetlosni potvrđni signal. Također, pokreće se servo motor koji predstavlja otključavanje vrata. Moguće je prilagoditi pristup za otključavanje vrata u aplikacijama koje zahtijevaju visoku sigurnost ovisno o mjeri pouzdanosti prepoznavanja otiska.

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

#define LED_PINZ A0
#define LED_PINC A1
#define BUZZER 4
#define LockMotor 6

SoftwareSerial mySerial(2, 3);
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup() {
    Serial.begin(9600);
    pinMode(LED_PINZ, OUTPUT);
    noTone(BUZZER);
    while (!Serial);
    delay(100);
    finger.begin(57600);
    if (finger.verifyPassword()) {
        Serial.println("Found fingerprint sensor :)");
    } else {
        Serial.println("Did not find fingerprint sensor :(");
        while (1) { delay(1); }
    }
    finger.getTemplateCount();
    Serial.print("Sensor contains ");
    Serial.print(finger.templateCount);
    Serial.println(" templates");
    Serial.println("Waiting for valid finger...");
}

void loop() {
    getFingerprintIDez();
    delay(50);
}

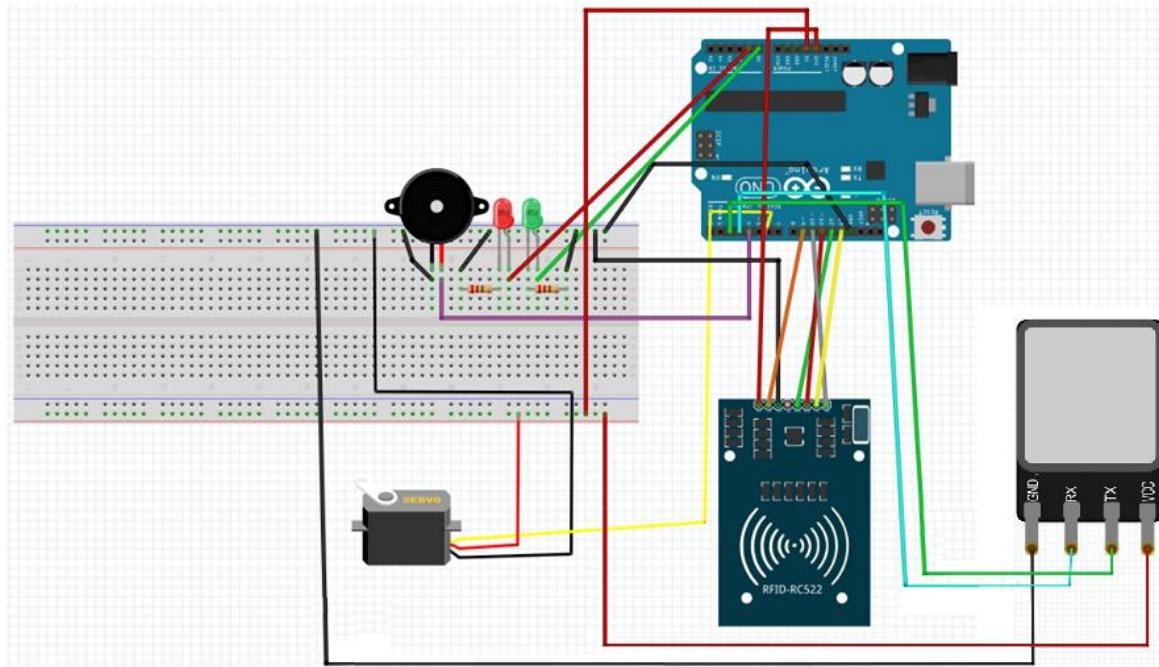
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;
    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;
    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // found a match!
    Serial.print("Found ID #");
    Serial.print(finger.fingerID);
    Serial.print(" with confidence of ");
    Serial.println(finger.confidence);
    digitalWrite(LED_PINZ, HIGH);
    tone(BUZZER, 500);
    doorUnlock();
    delay(2000);
    noTone(BUZZER);
    doorLock();
    digitalWrite(LED_PINZ, LOW);
    return finger.fingerID;
}
```

## Podsustav RFID i otisak prsta

Slika 11 prikazuje podsustav koji objedinjuje prepoznavanje korisnika preko RFID-a, otiska prsta i omogućuje otključavanje vrata na način opisan u prethodnim poglavljima. Analogno je moguće ostvariti i otvaranje vrata, korištenjem motora s reduktorom, što se nije ostvarilo jer Arduino

UNO nema dovoljan broj izvoda za takav zahtjev. Problem bi se riješio upotrebom Arduino MEGA pločice.



Slika 11: Podsustav za upravljanje otključavanja vrata

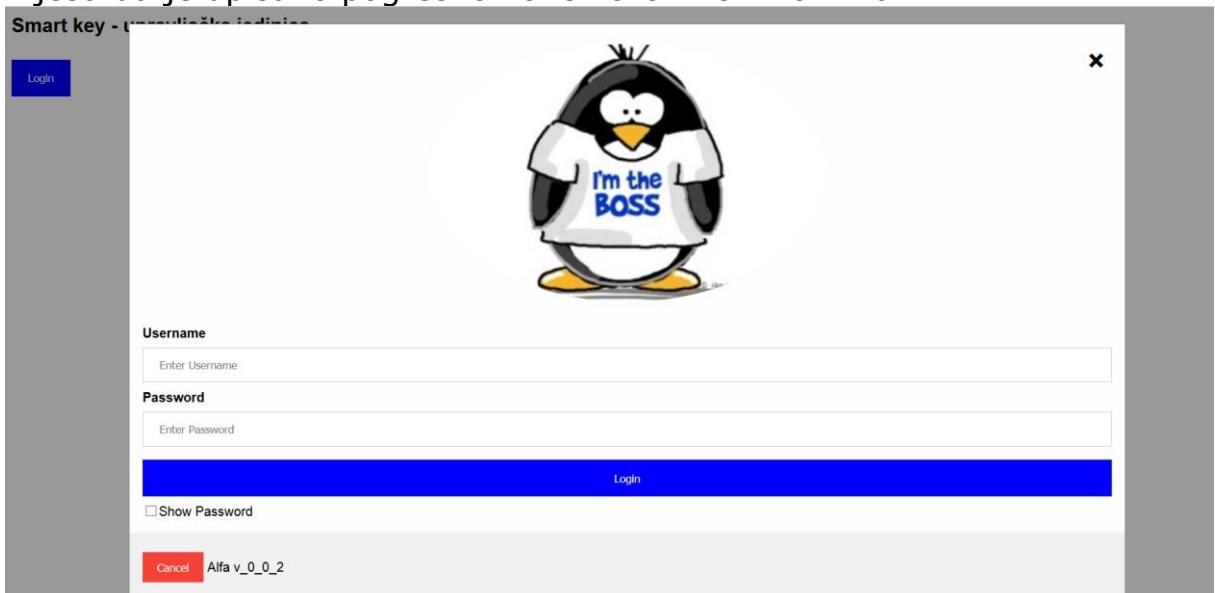
### 3.3. Prepoznavanje korisnika preko web servisa (Moštak)

Web upravljačka jednica za Smart key sustav napisana je u PHP-u. PHP je open source programski jezik namijenjen za izradu dinamičnih web stranica.

Prilikom inicijalnog pristupa na web stranicu prikazuje se gumb „Login“ (slika 12.).

**Smart key - upravljačka jedinica****Slika 12. inicijalni prikaz na web stranici**

Pritiskom na gumb „Login“ otvara se modalni prozor u kojem se traži od korisnika da unese korisničko ime i lozinku (slika 13.). Unutar modalnog prozora moguće je koristiti funkcionalnost prikaza lozinke. Nakon unosa podataka u slučaju neispravnog korisničkog imena ili lozinke korisnika se preusmjerava na inicijalni prikaz, no ovaj put uz prikaz obavijesti da je upisano pogrešno korisničko ime ili lozinka.



The image shows a login modal window titled "Smart key - upravljačka jedinica". It features a blue "Login" button at the top left. In the center is a cartoon penguin wearing a white t-shirt that says "I'm the BOSS". Below the penguin are two input fields: "Username" and "Password", each with a placeholder "Enter Username" and "Enter Password" respectively. To the right of the password field is a "Show Password" checkbox. At the bottom of the modal are "Cancel" and "Alfa v\_0\_0\_2" buttons, with the "Alfa v\_0\_0\_2" button being red. The background of the modal is white, and it has a close button "x" in the top right corner.

**Slika 13. Obrazac za login u korisničku bazu podataka**

Nakon uspješnog ulaska u upravljačku jedinicu unošenjem ispravnog korisničkog imena i lozinke, na zaslonu se pojavljuje ispis trenutne baze korisnika, kao i trenutno omogućene modifikacije baze-dodavanje novog korisnika ili brisanje postojećih (slika 14.) (programska podrška za

modifikaciju trenutnih korisnika u vidu mijenjanja vremenskog intervala dopuštenog ulaza ili dozvole je još uvijek u izradi).

The screenshot shows a web application interface titled "Korisnička baza". At the top right is a "Logout" button. On the left, a blue button says "Stvor novog korisnika". The main area displays a table with columns: "Id", "Dozvola", "Vrijeme ulaz (od)", and "Vrijeme ulaz (do)". The data in the table is as follows:

Id	Dozvola	Vrijeme ulaz (od)	Vrijeme ulaz (do)
1 1		12:00:00	14:00:00
2 1		00:00:00	23:59:59
3 0		00:00:00	00:00:00
4 1		12:00:00	14:00:00
5 1		08:00:00	14:00:00
6 0		00:00:00	00:00:00
7 1		10:00:00	16:00:00
8 1		00:00:00	23:59:59
9 1		00:00:00	23:59:59
10 1		08:00:00	16:00:00

Below the table is a blue "DELETE" button.

Slika 14. Web upravljačka jedinica

Novi korisnik se dodaje pritiskom na „Stvor novog korisnika“. Pritiskom se otvara novi modalni prozor za unos podataka (slika 15.)

The screenshot shows a modal dialog box titled "Korisnička baza". It contains a table with columns: "Id", "Dozvola", "Vrijeme ulaz (od)", and "Vrijeme ulaz (do)". Below the table are input fields: "id" (with placeholder "id"), "Dozvola pristupa" (dropdown menu), "Vrijeme od kada je dozvoljen ulaz: (pr. 08:00:00)" (time input field), and "Vrijeme do kada je dozvoljen ulaz: (pr. 16:00:00)" (time input field). At the bottom are two buttons: "Stvor novog korisnika" (blue) and "Cancel" (red).

Slika 15. Dodavanje novog korisnika

Ručno se moraju odabrati sve karakteristike za novog korisnika, uključujući njegov jedinstveni ID i podatak ima li uopće dozvoljen pristup (ovo se može koristiti ako se nekome trenutno želi namjerno zabraniti pristup). Ako mu je pristup dozvoljen, potrebno je unijeti vremenski interval od kada do kada mu je dozvoljen pristup u sustav, tj. od kada do kada korisnik može upravljati i otključavati vrata. Cjelokupni programski kod je dostupan u prilogu kod.zip, a u nastavku je prikaz datoteke „insert.php“ koja se poziva nakon pravilnog unosa podataka za unos novog korisnika u bazu i pritiskom na gumb „Unesi novog korisnika“.

```
<?php
```

```

if(isset($_POST['id'])){
    $id=$_POST['id'];
}
if(isset($_POST['dozvola'])){
    $dozvola=$_POST['dozvola'];
}
if(isset($_POST['vrijeme_ulaz'])){
    $vrijeme_ulaz=$_POST['vrijeme_ulaz'];
}
if(isset($_POST['vrijeme_izlaz'])){
    $vrijeme_izlaz=$_POST['vrijeme_izlaz'];
}

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "smart key";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "INSERT INTO korisnici (id, dozvola, vrijeme_ulaz, vrijeme_izlaz)
VALUES ('$id', '$dozvola', '$vrijeme_ulaz', '$vrijeme_izlaz')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
header('Location: http://localhost/smartkey/db.php');
?>

```

Ovisno o browseru, unos podataka za vrijeme od kada do kada je dozvoljen ulaz biti će u drugačijem formatu (pr. u Firefoxu je u formatu 0-12 AM/PM, dok je u Chromeu u formatu 00-24).

Brisanje korisnika iz baze je vrlo jednostavno. S lijeve strane se označi koje korisnike se želi obrisati i potom se pritisne na „DELETE“.

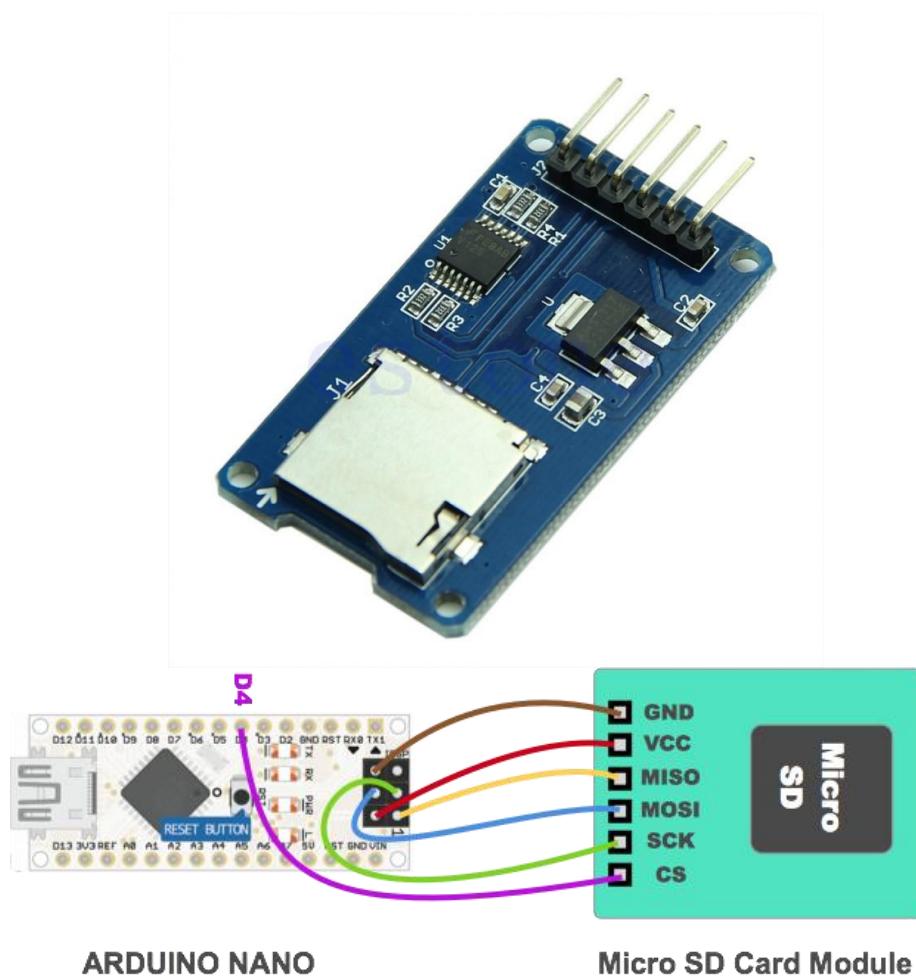
Nakon što je korisnik završio pregled i modifikaciju baze, potrebno se odlogirati pritiskom na gumb Logout.

U svrhu testiranja korišten je besplatni Wampserver (za windows operativni sustav) koji u sebi ima integriran Apache web server i MySQL servise.

### **3.4. Provjera korisnika pomoću SD kartice (Moštak)**

Micro SD kartica spojena je na Arduino NANO kako bi baza korisnika bila konstantno dostupna, neovisno o ostalim utjecajima na sustav. SD

Card Module (slika 16.) komunicira s arduinom preko sinkrone serijske komunikacije SPI. SPI zahtjeva 4 pina za kontrolu i sinkronizaciju i 2 pina za napajanje (slika 16.).



**Slika 16. micro SD kartica i način spajanja na Arduino nano**

Na arduino je osim microSD modula spojen i 7 segmentni LED prikaznik sa 8 znamenaka (MAX 7219), kako bi se mogli prikazati rezultati pretrage korisnika po bazi podataka.

Led prikaznikom se upravlja pomoću „LedControl library“. U slučaju da korisnik postoji i pristup mu je odobren, na prikazniku se ispisuje poruka „HELLO-id“, gdje je id identifikacija korisnika. U slučaju da je korisniku

zabranjen pristup, na prikazniku se ispisuje poruka „odbi—id“. Ako korisnik ne postoji u bazi podataka sustav ne reagira (moguća modifikacija je da sustav ispisuje „error-id“). Pregled odsječka programskega koda za povezivanje s microSD karticom i učitavanje baze podataka dan je u nastavku, kao i primjer funkcije za ispis na LED prikaznik ako je korisniku odobren ulaz:

```
Serial.print("Ucitavanje SD kartice...");

if (!SD.begin(SD_PIN)) {
    Serial.println("No SD-card.");
    return;
}

Serial.println("Ucitavanje uspjesno.");
if (SD.exists(db_name)) {

    dbFile = SD.open(db_name, FILE_WRITE);

    // Sometimes it wont open at first attempt, especially after cold start
    // Let's try one more time
    if (!dbFile) {
        dbFile = SD.open(db_name, FILE_WRITE);
    }

    if (dbFile) {
        Serial.print("Opening current table... ");
        EDB_Status result = db.open(0);
        if (result == EDB_OK) {
            Serial.println("DONE");
        }
    } else {
        Serial.println("Could not open file " + String(db_name));
        return;
    }
}

//primjer funkcije za ispis na LED
void ulazak_LED(int id){

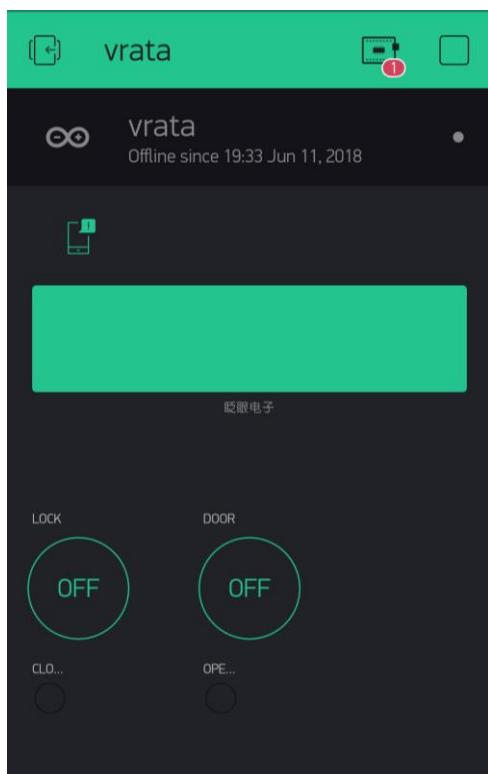
    lc.clearDisplay(0);
    lc.setChar(0,7,'H',false);
    lc.setChar(0,6,'E',false);
    lc.setChar(0,5,'L',false);
    lc.setChar(0,4,'L',false);
    lc.setChar(0,3,0,false);
    lc.setChar(0,2,'-',false);
    lc.setChar(0,1,id/10%10,false);
    lc.setChar(0,0,id%100,false);
    delay(2000);
    lc.clearDisplay(0);
    lc.setColumn(0,7,0xff);
}
```

### 3.5. Zadavanje uputa preko mobilne aplikacije (Bucić)

Mobilna aplikacija napravljena je korištenjem platforme Blynk koja omogućava jednostavnu komunikaciju između mobilnog uređaja i Arduina. Platforma se jednostavno koristi, a jednom napravljen projekt može se lagano dijeliti na druge uređaje.

Prilikom izrade novog projekta potrebno je definirati tip uređaja koji se koristi te način povezivanja. U ovom slučaju to su Arduino Uno i USB. Nakon kreiranja novog projekta putem e-maila dobiva se token koji je kopirati u kod kako bi se uređaj povezao s projektom.

Unutar aplikacije postoje predefinirani elementi koji odgovaraju fizičkim komponentama koje se uobičajeno koriste uz Arduino. Besplatna verzija aplikacije ograničava broj elemenata koji se mogu dodati u projekt. Primjer korisničkog sučelja dan je na Slika 17. U trenutnoj izvedbi elementi za ispis obavijesti se ne koriste, ali su dodani kao mogućnost za daljnju nadogradnju sustava. Također moguće je dodati i element za terminal koji bi onda omogućio razmjenu različitih tipova podataka i mogući unos novih korisnika u bazu preko aplikacije, no to je ostavljeno za sljedeću verziju sustava.



Slika 17: Upravljačko sučelje za mobitel

U nastavku su dani dijelovi koda koji se odnose na upravljanje korištenjem aplikacije. Kako bi se ostvarila komunikacija između uređaja

potrebno je uključiti pripadne programske biblioteke te SoftwareSerial. Kako je ranije spomenuto potrebno je uključiti i token koji odgovara projektu koji se povezuje na fizički sustav. Mana ovog modula je to što se token pridjeljuje projektu, a ne specifičnom uređaju, pa ne postoji način na koji bi se korisnici razlikovali.

```
#include <SoftwareSerial.h>

SoftwareSerial SwSerial(10, 11); // RX, TX

#include <BlynkSimpleStream.h>

// You should Auth Token in the Blynk App.
char auth[] = "5757a26623d9486aba8bc77b03cd9486";
```

Neki elementi u aplikaciji imaju mogućnost direktnog postavljanja digitalnih i analognih izlaza kada je potrebno samo jednostavno zadavanje naredbi. U ovom primjeru podatci se razmjenjuju preko varijabli koje se potom procesuiraju u kodu kako bi se dobile upravljačke vrijednosti za izvršni podsustav.

```
BLYNK_WRITE(V2) {
    blinkLockButtonVal = param.toInt();
    pinChangedLock = true;
}
```

```
WidgetLED led2(V4);
```

U glavnoj petlji pokreće se platforma Blynk, a potom se poziva funkcija u kojoj su smještene sve obrade podataka povezane s aplikacijom.

```
Blynk.run();
blynkValues();
```

U aplikaciji korisnik može zadati željeno stanje ključanice i vrata te mu je u isto vrijeme omogućen nadzor nad položajem, odnosno prikazuje se jesu li vrata u nekoj od krajnjih pozicija.

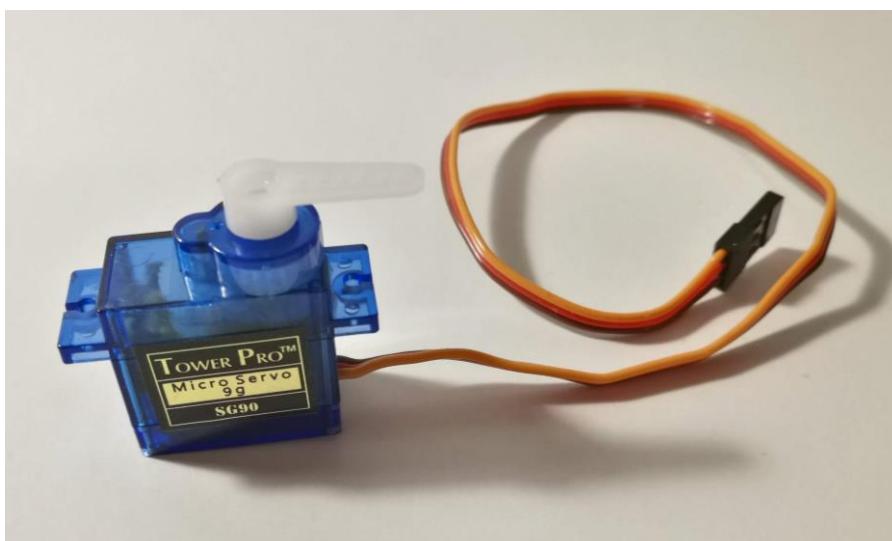
```
void blynkValues() {
    if (openHallChanged) {
        if (openHall) {
            led2.on();
        } else {
            led2.off();
        }
        openHallChanged = false;
    }

    if (pinChangedLock) {
        if (blinkLockButtonVal == 1 && lockHall==1) {
            lock=0;
        }
    }
}
```

```
    } else if (blinkLockButtonVal == 0 && lockHall==1) {  
        lock=1;  
    }  
    pinChangedLock = false;  
}  
...  
...
```

### 3.6. Otključavanje i zaključavanje (Bucić)

Za demonstraciju upravljanja bravom koristi se mali servo motor prikazan na Slika 18: Servo motor. U slučaju kada se vrata i stvarno mogu lagano otključati okretanjem ključanice samo je potrebno prikladno pričvrstiti motor.



Slika 18: Servo motor

U programskom kodu implementirano je zakretanje motora za 180 stupnjeva te motor može biti u poziciji 0 ili 180 stupnjeva koje predstavljaju da je brava otključana, odnosno zaključana. U nastavku je dan programski kod funkcija za upravljanje bravom.

```
void doorUnlock(){ // goes from 0 degrees to 180 degrees  
    for (pos = 0; pos <= 180; pos += 1) { // in steps of 1 degree  
        t2=millis();  
        if((t2-t1)>15){  
            myservo.write(pos); // tell servo to go to position in variable 'pos'  
            t1=millis(); // waits 15ms for the servo to reach the position  
        }  
    }  
    t1=0;  
}  
  
void doorLock(){ // goes from 0 degrees to 180 degrees
```

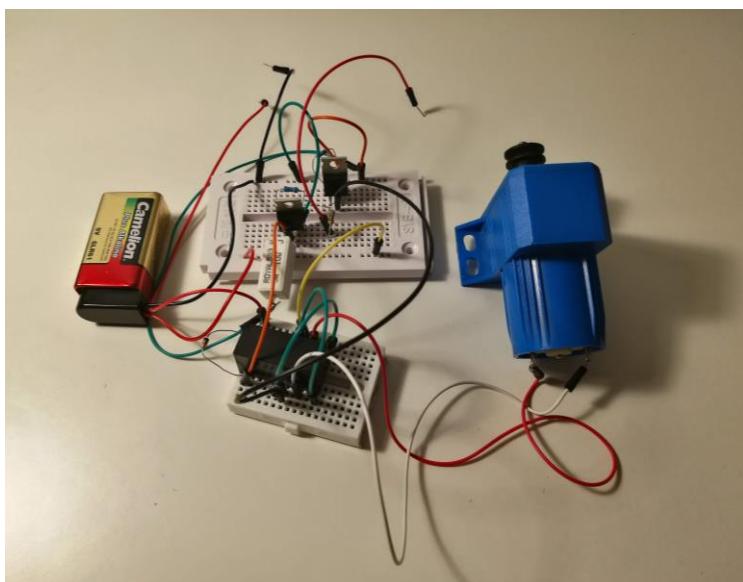
```
for (pos = 180; pos >= 0; pos -= 1) {      // in steps of 1 degree
    t2=millis();
    if((t2-t1)>15){
        myservo.write(pos);    // tell servo to go to position in variable 'pos'
        t1=millis(); // waits 15ms for the servo to reach the position
    }
    t1=0;
}
```

### 3.7. Otvaranje i zatvaranje (Bucić)

Otvaranje vrata kompleksniji je zadatak od otključavanja te zahtjeva više snage pa se zato koristi DC motor. Uz to, korišten je motor s reduktorom kako bi se uz istu snagu dobio veći moment. Za razliku od servo motora, u ovom slučaju nije moguće direktno spajanje na Arduino već je bilo potrebno koristiti dodatnu elektroniku.

Korišteni DC motor koristi približnu struju od 100 mA te mu je potrebno dovesti vanjsko napajanje. Osim zbog limita struje, koji kod Arduino Uno pločice iznosi oko 40 mA, motor je potrebno odvojiti u poseban strujni krug zbog toga što tijekom rada stvara šum koji bi utjecao na rad senzora.

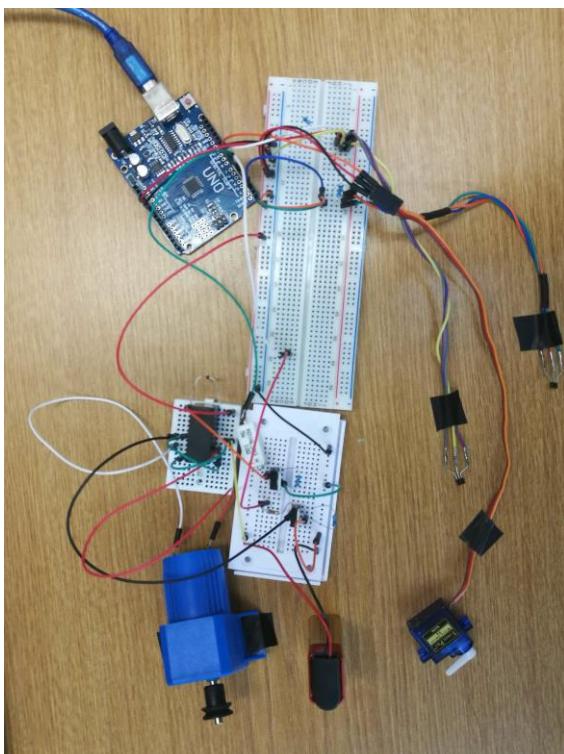
Dodatne komponente koje su korištene za upravljanje motorom su MOSFET tranzistori, otpornici, relej i dioda. Komponente su složene tako da čine dvije sklopke i preklopnik, pri čemu jedna sklopka pali i gasi motor, a druga upravlja relejom koji mijenja smjer vrtnje. Svaku sklopku čine jedan MOSFET [4] te dva otpornika, jedan od  $10\Omega$  preko kojeg se Arduino spaja na upravljačku elektrodu (*engl. gate*) MOSFET-a i jedan od  $1k\Omega$  koji se nalazi između uvoda (*engl. source*) i odvoda (*engl. drain*). Modul za otvaranje vrata prikazan je na Slika 19.



Slika 19: Modul za otvaranje vrata

Opisano upravljanje korištenjem tranzistora primjenjivo je samo za manje motore. Ako bi se u stvarnom sustavu koristio DC motor, za njegovo upravljanje bilo bi potrebno koristiti H-most. Više o upravljanju DC motorima može se pronaći na [5].

Kako bi se odredio položaj vrata dodana su dva Hall senzora [3]. Hall senzore je potrebno postaviti na rubne pozicije dok sama vrata moraju imati ugrađen magnet koji će se detektirati kada vrata dođu u krajnji položaj. Cijeli podsustav za upravljanje vratima dan je slikom Slika .



**Slika 20: Podsustav za upravljanje vratima**

U nastavku je dan kod koji demonstrira rad podsustava za otključavanje i otvaranje vrata. Inicijalna pozicija postavljena je tako da su vrata zatvorena i zaključana te je zatraženo otključavanje. Implementirana sekvenca kreće iz početnog stanja, otključava vrata, otvara ih, čeka 3 sekunde te ih potom zatvara i zaključava. Vrijednosti sa senzora dobivaju se pomoću prekida koji se pozivaju na promjenu vrijednosti.

```
#define CloseDoorHall 2
#define OpenDoorHall 3
#define DoorMotorOn 6
#define DoorMotorDirection 7
#define LockMotor 9
#define doorOpenTime 3000

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int lock=1; //if locked wanted 1, if unlocked wanted 0
int doorOpen=0;
int doorClose=0;
```

```
int pos = 0;      // variable to store the servo position
// We make these values volatile, as they are used in interrupt context
volatile bool lockHallChanged = false;
volatile int lockHall    = 0;
volatile int openHall=0;
volatile bool openHallChanged = false;
//time variables for delays between actions
long int t1=0;
long int t2=0;
long int t1door=0;
long int t2door=0;

// Storing the value and process it in the main loop.
void checkHallClosed()
{
    // Invert state
    lockHall = !digitalRead(CloseDoorHall);
    // Mark pin value changed
    lockHallChanged = true;
}
void checkHallOpened()
{
    openHall = !digitalRead(OpenDoorHall);
    openHallChanged = true;
}

void doorUnlock(){ // goes from 0 degrees to 180 degrees
    for (pos = 0; pos <= 180; pos += 1) { // in steps of 1 degree
        t2=millis();
        if((t2-t1)>15){
            myservo.write(pos); // tell servo to go to position in variable 'pos'
            t1=millis(); // waits 15ms for the servo to reach the position
        }
    }
    t1=0;
}

void doorLock(){ // goes from 0 degrees to 180 degrees
    for (pos = 180; pos >= 0; pos -= 1) { // in steps of 1 degree
        t2=millis();
        if((t2-t1)>15){
            myservo.write(pos); // tell servo to go to position in variable 'pos'
            t1=millis(); // waits 15ms for the servo to reach the position
        }
    }
    t1=0;
}

void setup()
{
    Serial.begin(9600);

    attachInterrupt(digitalPinToInterrupt(CloseDoorHall), checkHallClosed,
CHANGE);
    attachInterrupt(digitalPinToInterrupt(OpenDoorHall), checkHallOpened,
CHANGE);
    pinMode(DoorMotorDirection, OUTPUT);
    pinMode(DoorMotorOn, OUTPUT);
    digitalWrite(DoorMotorOn, LOW); //make sure that the motor is off
    myservo.attach(LockMotor); // attaches the servo on pin 9 to the servo
object
    //initialize locked state and set wanted to unlocked
    myservo.write(180);
```

```
lock=0;
}

void loop()
{
    //if doors are closed process the wanted lock state
    if (lock==0 && lockHall==1){
        doorUnlock();
        delay(1000);
        doorOpen=1;
    }else if (lock==1 && lockHall==1){
        doorLock();
    }

    if (doorOpen==1 && lock==0 && openHall!=1){
        digitalWrite(DoorMotorDirection, HIGH); //set the direction of a motor
        delay(10); //to make sure that direction has changed
        digitalWrite(DoorMotorOn, HIGH); //turn on the motor
    }
    if (doorOpen==1 && openHall==1){
        digitalWrite(DoorMotorOn, LOW);
        t1door=millis();
        doorOpen=0;
    }

    //delay between opening and closing doors
    if (doorOpen==0 && openHall==1){
        t2door=millis();
        if (t2door-t1door>=doorOpenTime){
            doorClose=1;
        }
    }

    if (doorClose==1 && lockHall!=1){
        digitalWrite(DoorMotorDirection, LOW);
        delay(10);
        digitalWrite(DoorMotorOn, HIGH);
    }
    if (doorClose==1 && lockHall==1 ){
        digitalWrite(DoorMotorOn, LOW);
        doorClose=0;
        delay(1000);
        lock=1;
    }
}
```

## 4. Zaključak

Opisani sustav donosi poboljšanja u vidu sigurnosti prostora te služi kao asistent u kućanstvu. U tekstu su objašnjeni načini autorizacije korisnika odnosno registracija korisnika putem RFID-a, otiska prsta i mobilnog uređaja. Uz to objašnjeno je korištenje statičkih i dinamičkih baza koje bi omogućile upravljanjem dozvolom pristupa. Tako bi se putem jednostavnog web servisa omogućilo da ovlaštena osoba dozvoli ili zabrani pristup nekom korisniku niže razine. Kao sljedeći korak sustav bi se mogao lagano nadograditi da pamti i zapisuje korisnikovo točno vrijeme ulaska i izlaska iz prostorije. Sustav je zamišljen hijerarhijski i modularno te je stoga pogodan za različite prostore, od zaštite vlastite sobe u kući do ulaznih vrata. Osim iz aspekta sigurnosti sustav se može koristiti i kao pomoć u kućanstvu. Korisnik može udaljeno otključati i otvoriti, odnosno zatvoriti i zaključati vrata. Ta funkcionalnost posebno je korisna ljudima sa smanjenom pokretljivošću kojima otvaranje vrata predstavlja problem. Uz to, sustav je od pomoći u situacijama kada su korisniku pune ruke ili prenosi nešto pa bi mu otključavanje i otvaranje predstavljalo problem.

U ovom projektu dana je baza podsustava koji bi se trebao nadograditi ovisno o potrebama korisnika. Neke mogućnosti nadogradnje bile bi dodavanje dinamičkog vremena nakon kojeg se vrata otvaraju ili zatvaraju, a koje bi zadavao sam korisnik. Također potrebno je povezati sve podsustave u jednu cjelinu i implementirati na stvarni sustav.

## 5. Literatura

- [1] Arduino. URL: <https://www.arduino.cc/> (2018-06-11)
- [2] Blynk. URL: <https://www.blynk.cc/> (2018-06-11)
- [3] Hall senzor specifikacije URL: <http://www.alldatasheet.com/datasheet-pdf/pdf/345668/HONEYWELL/SS441R.html> (2018-06-12)
- [4] Mosfet specifikacije URL: <http://www.alldatasheet.com/datasheet-pdf/pdf/446095/RUICHIPS/RU6099R.html> (2018-06-12)
- [5] Nedelkovski, Dejan. Arduino DC Motor Control Tutorial – L298N | PWM | H-Bridge, URL: <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/> (2018-06-12)
- [6] Opalić, Milan. Prijenosnici snage i gibanja, URL: [https://hr.wikipedia.org/wiki/Mehani%C4%8Dki\\_prijenos](https://hr.wikipedia.org/wiki/Mehani%C4%8Dki_prijenos) (2018-06-13)
- [7] Osor promer, brave za sve tipove vrata, URL: <http://osor-promet.hr/proizvodi;brave-za-sve-tipove-vrata/> (2018-06-13)
- [8] Wampserver, URL: <http://www.wampserver.com/en/> (2018-06-14)
- [9] W3school, URL: <https://www.w3schools.com> (2018-06-14)

# Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
White paper	Kratak dokument koji daje uvid u neko područje, tehniku, politiku, proizvod, metodu, standard i sl.	<a href="https://en.wikipedia.org/wiki/White_paper">en.wikipedia.org/wiki/White_paper</a>
Arduino	Arduino razvojna platforma	<a href="https://www.arduino.cc/">https://www.arduino.cc/</a>
Blynk	Blynk razvojna platforma	<a href="https://www.blynk.cc/">https://www.blynk.cc/</a>
Servo motor	Rotacijski ili linearni aktuator	<a href="https://en.wikipedia.org/wiki/Servomotor">https://en.wikipedia.org/wiki/Servomotor</a>
DC motor	Istosmjerni motor	<a href="https://en.wikipedia.org/wiki/DC_motor">https://en.wikipedia.org/wiki/DC_motor</a>
Relej	Sklopni aparat	<a href="https://en.wikipedia.org/wiki/Relay">https://en.wikipedia.org/wiki/Relay</a>
MOSFET tranzistor	Aktivni poluvodički element	<a href="https://en.wikipedia.org/wiki/MOSFET">https://en.wikipedia.org/wiki/MOSFET</a>
Hall senzor	Senzor pozicije	<a href="https://en.wikipedia.org/wiki/Hall_effect_sensor">https://en.wikipedia.org/wiki/Hall_effect_sensor</a>
Vratilo	Pogonska osovina	<a href="https://hr.wikipedia.org/wiki/Vratilo">https://hr.wikipedia.org/wiki/Vratilo</a>
Mehanički prijenos	Prijenos snage ili gibanja uz pomoć mehaničkog sklopa ili prijenosnika	<a href="https://hr.wikipedia.org/wiki/Mehani%C4%8Dki_prijenos">https://hr.wikipedia.org/wiki/Mehani%C4%8Dki_prijenos</a>
RFID	Razmjena podataka pomoću radio frekvencije	<a href="https://hr.wikipedia.org/wiki/RFID">https://hr.wikipedia.org/wiki/RFID</a>



## Smart key

