

# Model protu-provalnog sustava temeljen na Arduino platformi

---

**Radočaj, Dražen**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:913589>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-07-10**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

**MODEL PROTU-PROVALNOG SUSTAVA TEMELJEN NA  
ARDUINO PLATFORMI**

Rijeka, rujan 2023.

Dražen Radočaj

0069079981

SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

**MODEL PROTU-PROVALNOG SUSTAVA TEMELJEN NA  
ARDUINO PLATFORMI**

Mentor: doc. dr. sc. Ivan Volarić

Rijeka, rujan 2023.

Dražen Radočaj

0069079981

Rijeka, 20. ožujka 2023.

Zavod: **Zavod za automatiku i elektroniku**  
Predmet: **Digitalna elektronika**  
Grana: **2.03.03 elektronika**

## ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Dražen Radočaj (0069079981)**  
Studij: **Sveučilišni prijediplomski studij elektrotehnike**

Zadatak: **Model protu-provalnog sustava temeljen na Arduino platformi / Arduino based anti-burglar system model**

### Opis zadatka:

U sklopu završnog rada potrebno je izraditi model protu-provalnog sustava. Model se sastoji od modela vrata, po čijem se otvaranju uključuje alarmno stanje, kojeg korisnik može isključiti na tri načina: unosom šifre, RFID karticom ili pomoću otiska prsta. U slučaju neuspješnog isključivanja alarmnog stanja generira se zvučni signal koji upozorava na provalu. Senzor otvorenosti vrata, tipkovnicu, te čitače RFID kartice i otiska prsta potrebno je spojiti na Arduino razvojnu pločicu, te napisati odgovarajuću programsku podršku. Osim toga, potrebno je implementirati mogućnost logiranja, tj. zapisivanja vremena u kojima se alarm uključio i isključio.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.



Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:



---

Doc. dr. sc. Ivan Volarić

Predsjednik povjerenstva za  
završni ispit:



---

Prof. dr. sc. Dubravko Franković

## **IZJAVA**

Ja Dražen Radočaj izjavljujem kako sam samostalno izradio završni rad na temu „Model protuprovalnog sustava temeljen na Arduino platformi“

Potpis:

---



# Sadržaj

<b>1. UVOD</b> .....	1
<b>2. ARDUINO</b> .....	2
<b>3. KORIŠTENI MODULI</b> .....	5
<b>3.1 Senzor otiska prsta</b> .....	5
<b>3.2 RFID senzor</b> .....	6
<b>3.3 RTC modul</b> .....	7
<b>3.4 Modul za SD karticu</b> .....	8
<b>4. SCHEMA SPOJA</b> .....	9
<b>4.1. Shema spoja kruga napajanja</b> .....	9
<b>4.3 Shema spoja razvojne pločice Arduino Mega</b> .....	12
<b>4.4 Shema spoja razvojne pločice Arduino Nano</b> .....	13
<b>5. PROGRAMSKA PODRŠKA</b> .....	14
<b>6. 3D PRINTANJE TE IZRADA MODELA U PROGRAMU CATIA</b> .....	24
<b>6.1. CATIA</b> .....	24
<b>6.2. 3D printanje</b> .....	28
<b>7. KONAČAN IZGLED RADA SIGURNOSNOG SUSTAVA</b> .....	30
<b>8. ZAKLJUČAK</b> .....	31
<b>9. LITERATURA</b> .....	32
<b>10. SAŽETAK I KLJUČNE RIJEČI</b> .....	33
<b>11. ABSTRACT AND KEY WORDS</b> .....	34

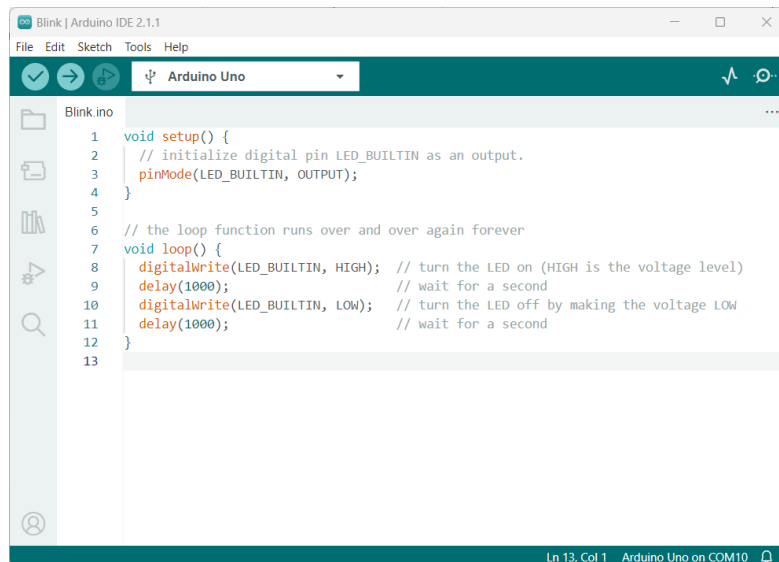
# 1. UVOD

Zbog učestalosti provala razvio se problem za povećanjem sigurnosti. Područje elektronike ponudilo je rješenje koristeći se senzorima te mikrokontrolerima kako bi se otkrile neželjene radnje te poduzele sigurnosne mjere kao što su zvučna, svjetlosna ili neka druga signalizacija. U sklopu ovog završnog rada izrađen je model protu-provalnog sustava putem Arduino platforme. Arduino prima signal sa svjetlosnog senzora koji detektira da li su vrata otvorena ili zatvorena. Ukoliko je alarmno stanje uključeno te se vrata otvore, oglasi se zvučni signal koji upozorava da je potrebno unijeti lozinku odnosno prisloniti RFID karticu ili otisak prsta kako bi se alarmno stanje isključilo. Ukoliko se alarmno stanje ne isključi u predefiniranom vremenskom intervalu oglašava se dodatan zvučni signal koji obavještava da je došlo do provale. Unosom lozinke, u alarmnom stanju, korisnik ima 3 pokušaja kako bi se smanjila vjerojatnost pokušaja pogađanja lozinke. Ukoliko to ne uspije također dolazi do oglašavanja dodatnog zvučnog signala. U slučaju da korisnik nije uspio unijeti ispravnu lozinku unutar intervala, alarmno stanje se može isključiti samo otiskom prsta. Svako uključivanje te isključivanje alarmnog stanja se zapisuje na micro SD memorijsku karticu na kojoj se bilježi vrijeme promjene stanja, stanje koje se postavilo te metoda promjene stanja kojom se korisnik koristio (lozinka, RFID, otisak prsta).



## 2. ARDUINO

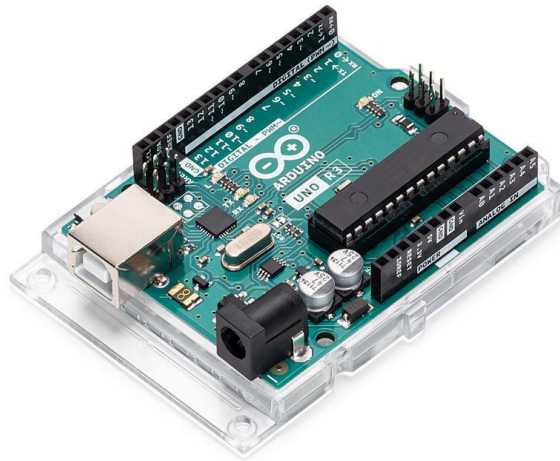
Arduino je tvrtka koja nudi jednostavno programiranje svojih razvojnih pločica putem C programskog jezika. Arduino je „open-source“ što znači da je korisnicima lako dostupan materijal za učenje te zbog međusobne suradnje među korisnicima, dostupni su i programi odnosno kodovi koji se mogu koristiti u rješavanju sličnih problema. Također je dostupan velik izbor biblioteka koje se unutar programa mogu koristiti za lako korištenje raznih modula, senzora, itd. Programi se pišu u Arduino-vom razvojnom sučelju koje se zove Arduino IDE (eng. *Arduino Integrated Development Environment*), prikazanog na slici 2.1.



```
Blink.ino
1 void setup() {
2   // initialize digital pin LED_BUILTIN as an output.
3   pinMode(LED_BUILTIN, OUTPUT);
4 }
5
6 // the loop function runs over and over again forever
7 void loop() {
8   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
9   delay(1000); // wait for a second
10  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
11  delay(1000); // wait for a second
12 }
13
```

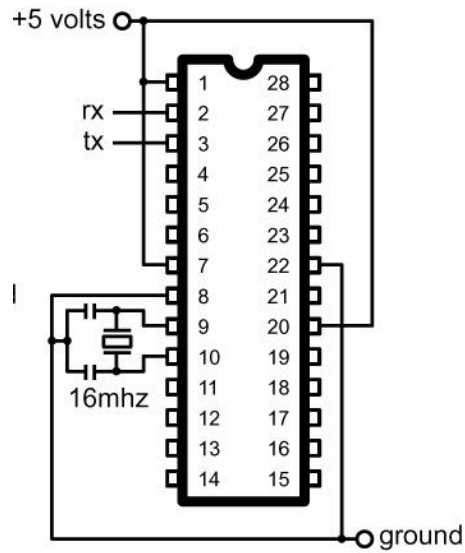
Slika 2.1 Arduino IDE razvojno sučelje.

Razvojno sučelje Arduino IDE, osim što služi za prenošenje programa na Arduino pločicu, ima i razne druge mogućnosti kao što je serial monitor koji služi za praćenje serijske komunikacije između Arduino pločice i računala, koja se odvija putem RX i TX pinova Arduina. Arduino za svoje pločice koristi Atmelove mikrokontrolere. Najčešće korišten Arduino Uno, prikazan na slici 2.2, koristi mikrokontroler ATmega 328p.



*Slika 2.2. Razvojna pločica Arduino Uno [1]*

Brzina radnog takta mikrokontrolera je 16MHz, te ima 2KB ram memorije i 32KB Flash memorije u kojoj je program spremljen. Razvojna pločica Arduino Uno nudi 12 digitalnih pinova, koji se mogu koristiti kao ulazni ili izlazni pinovi, te 5 analognih pinova koji se koriste isključivo kao ulazni pinovi za očitavanje analognih vrijednosti. Arduino pločice rade na naponu od 5V. Analogni ulazi su spojeni na analogno-digitalni pretvornik rezolucije 10 bita pa može razlikovati 1024 vrijednosti napona signala, stoga ima osjetljivost od 4,89mV. Neki od digitalnih pinova se mogu koristiti kao analogni izlazi pomoću pulsno širinske modulacije rezolucije 8 bita i frekvencije 500Hz. Razvojna pločica Arduino Mega je unaprijeđena verzija pločice Arduino Uno, koja koristi mikrokontroler ATmega 2560 te se češće koristi za sklopove gdje je potrebna veća stabilnost sustava te ukoliko se javlja potreba za korištenje više analognih ili digitalnih pinova. Arduino Mega pločica ima 54 digitalna pina te 16 analognih pinova. Brzina radnog takta mikrokontrolera je 16MHz te ima 8KB radne memorije i 256KB Flash memorije. Još jedna od dosta često korištenih Arduino pločica je Arduino Nano. Ova pločica je male površine te su pinovi pločice prilagođeni za DIL podnožje te je najpogodnija za ugradnju na tiskanu pločicu. Arduino Nano koristi isti mikrokontroler kao i Arduino Uno. Mikrokontroler na Arduino razvojnim pločicama dolazi s već isprogramiranim *bootloader*-om, koji omogućuje programiranje mikrokontrolera putem serijske komunikacije, koja je ostvarena preko USB priključka. Također je moguće i klasično programirati mikrokontroler, preko ICSP pinova. Mikrokontroler može raditi i samostalno, bez Arduino razvojne pločice, spajanjem oscilacijskog kruga izvedenog pomoću kristalnog oscilatora te dva kondenzatora, kao što je to prikazano na slici 2.3.



*Slika 2.3. Shema spajanja oscilacijskog kruga mikrokontrolera ATmega 328p [2]*

Osim razvojnih pločica, Arduino nudi i Arduino *shield*-ove prikazane na slici 2.4. To su pločice koje se jednostavno postavljaju na Arduino pločicu kao što su Arduino Uno i Mega, te pružaju dodatne mogućnosti kao što su GPS, GSM, ethernet, itd.

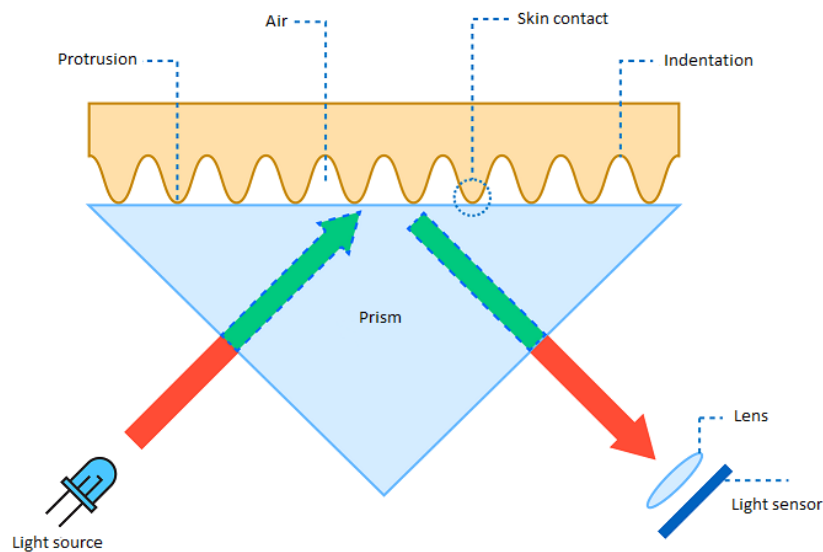


*Slika 2.4. Arduino shield pločice [3]*

### 3. KORIŠTENI MODULI

#### 3.1 Senzor otiska prsta

Senzor otiska prsta spada pod optičke senzore. Kako se prst prislanja na senzor, zrake vanjske svjetlosti ne mogu doći između prsti i senzora stoga je u senzor ugrađen izvor svjetlosti (LED dioda). Kako bi se točnije otkrile linije otiska prsta, izvor svjetlosti je postavljen tako da se zrake svjetlosti, kroz prizmu, odbijaju pod nagibom te kroz leću upadaju na svjetlosni senzor, kao što je to prikazano na slici 3.1.



Slika 3.1. Princip rada senzora otiska prsta [4]

Zrake svjetlosti, zbog različite udaljenosti te kuta refleksije između linija otiska prsta te dodirne površine senzora, različito upadaju kroz leću te tako senzor izrađuje sliku otiska prsta. Senzor otiska prsta DY50, prikazan na slici 3.2. može spremiti do 300 različitih slika otiska prsta. Usporedbom novo izmjerene slike otiska prsta, sa slikama iz spremljene baze podataka, moguće je identificirati otisak. Senzor otiska prsta DY50 radi na naponu od 3,3V. Sigurnosna razina ovog senzora je 3 od maksimalne razine 5.



Slika 3.2. Senzor otiska prsta DY50 [5]

### 3.2 RFID senzor

RFID (*Radio-frequency identification*) senzori najčešće koriste frekvencije 125kHz i 134kHz te je maksimalna udaljenost za ostvarivanje komunikacije do 10cm. Da bi se uspješno ostvarila komunikacija potrebno je imati dvije antene: za odašiljanje i primanje signala. To je najčešće izvedeno putem zavojnice odnosno spirale koja se nalazi na tiskanoj pločici RFID senzora te prijenosnih uređaja („tagova“) poput kartice, naljepnice ili privjeska itd. Najčešće izvedbe prijenosnih uređaja su pasivni elementi što znači da prilikom prislanjanja uređaja pored senzora, na spiralnoj zavojnici se inducira napon zbog promjene elektromagnetskog toka:

$$e = -N * \frac{d\theta}{dt}, \quad (3.1)$$

gdje je  $N$  broj namota zavojnice,  $d\theta$  je promjena elektromagnetskog toka dok je  $dt$  vrijeme u kojemu bila promjena toka. Unutar RFID prijenosnog uređaja se nalazi Flash memorija, dok RFID senzor generira RF signale. RFID senzor unutar sebe ima generator RF (radio frekvencijskog) signala te detektor RF signala. Kada generirani RF signal dođe do prijenosnog uređaja, inducirani napon napaja isti uređaj, koji potom, kroz pojačalo signala, šalje informacije istom metodom do RF senzora. Izgled RFID senzora i tagova prikazan je na slici 3.3.



*Slika 3.3. RFID senzor te prijenosni uređaji (tagovi) [6]*

### **3.3 RTC modul**

RTC (eng. *Real Time Clock*) se koristi za dobivanje točne informacije o vremenu. Koristi kristalni oscilator od 32kHz koji se nalazi unutar mikrokontrolera da bi se smanjila osjetljivost na vanjske utjecaje poput temperature. Na modul je ugrađena baterija kako bi modul nastavio mjeriti vrijeme i nakon isključivanja vanjskog napona napajanja. Izgled RTC modula je prikazan na slici 3.4.

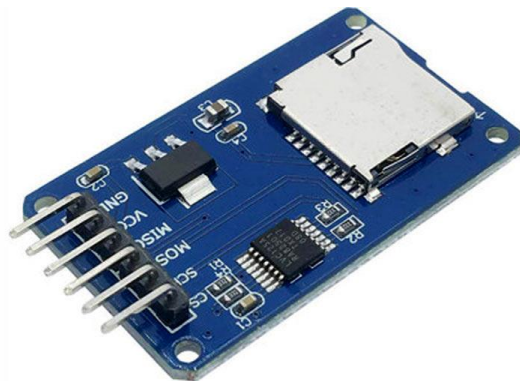


*Slika 3.4. RTC modul [7]*

RTC modul koristi IIC komunikaciju (eng. *Inter-Integrated Circuit*). Ovaj tip komunikacije sastoji se od dvije linije SDA (eng. *serial data*) te SCL (eng. *serial clock*). Putem SCL se šalje radni takt promjenjive frekvencije koji određuje brzina slanja bitova serijske komunikacije. Putem SDA se šalju informacije. Korištenjem ovog tipa komunikacije omogućeno je spajanje više uređaja korištenjem samo ove dvije linije, pošto svaki uređaj na IIC sabirnici ima svoju adresu. Svaki uređaj se razlikuje po svojoj adresi. Za ovu komunikaciju potrebno je odrediti glavni upravljački uređaj (eng. *master*) te sporedne uređaje (eng. *slave*).

### 3.4 Čitač za SD karticu

Čitač za SD karticu se koristi kako bi se trajno mogli zapisati podatci na micro SD memorijsku karticu. Osim zapisivanja, moguće je i čitanje spremljenih podataka. Izgled ovog čitača prikazan je na slici 3.5.



Slika 3.5. Čitač SD kartice [8]

Napon napajanja micro SD kartice iznosi 3,3V no zbog ugrađenog regulatora napona unutar ovog modula, moguće je napajati ovaj modul s izvorom napona od 5V. Modul SD kartice koristi SPI (eng. *Serial Peripheral Interface*) komunikaciju. Kao i kod IIC komunikacije koristi se linija s promjenjivom frekvencijom radnog takta (SCK) no ulazna i izlazna linija su odvojene pa imamo još dvije linije: MOSI (eng. *Master Output Slave Input*) i MISO (eng. *Master Input Slave Output*).





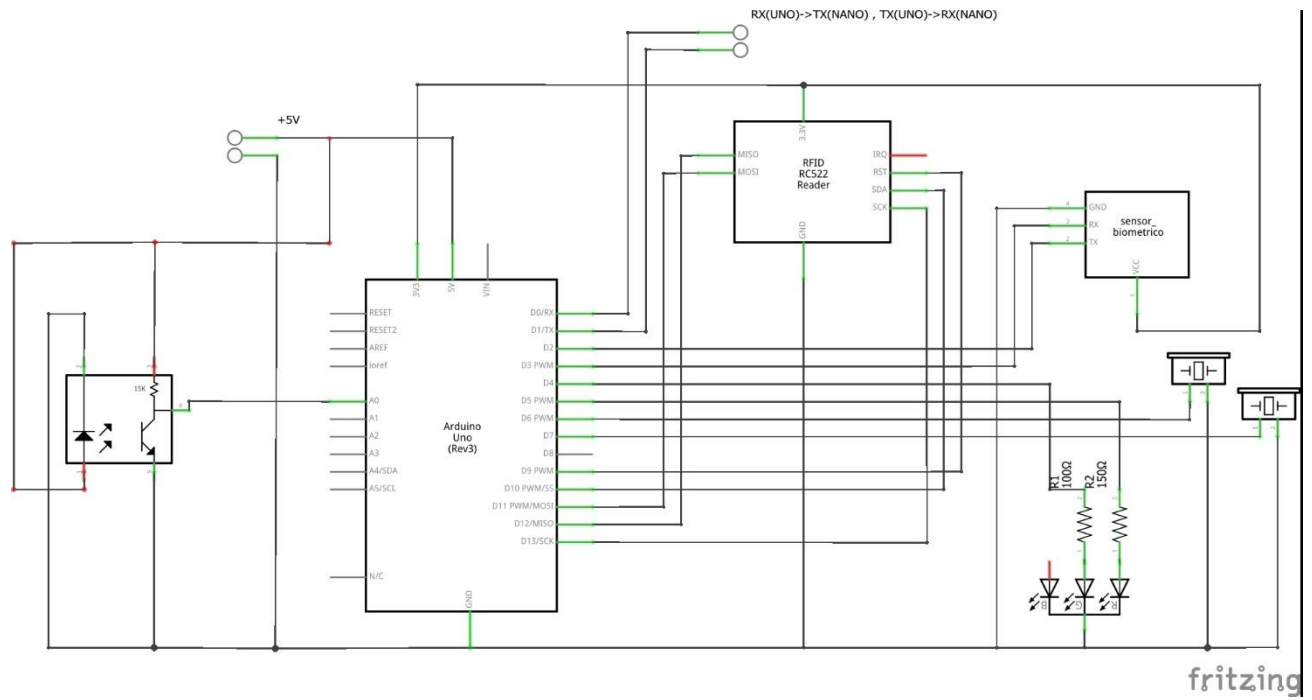


## 4.2 Shema spoja razvojne pločice Arduino Uno

Arduino Uno je korišten kao glavni upravljački element sklopa sigurnosnog sustava. Na njega su spojeni svi senzori, te elementi svjetlosne i zvučne signalizacije. Arduino ima ugrađen stabilizator napona LM6206 koji ulazni napon od 5V spušta na iznos od 3,3V te time nudi mogućnost izvora napajanja za neke od senzora kojima je potreban napon od 3,3V. Arduino putem senzora otvorenosti vrata (principa rada sličnog optokapleru) očitava da li su vrata otvorena ili zatvorena. Senzor je optički, te se sastoji od infracrvene LED diode, te foto-tranzistora. Prekidanjem svjetlosnog snopa između LED diode i foto-tranzistora, foto-tranzistor prestaje voditi struju. Mjerenjem napona na kolektoru tranzistora može odrediti da li je došlo do prekida svjetlosnog snopa. Taj napon se dovodi na analogni ulaz A0 Arduino razvojne pločice. Arduino putem RGB LED diode daje informaciju o stanju zaključanosti sigurnosnog sustava. Slanjem izlaznog napona na digitalne izlaze uključuje odnosno isključuje crveni i zeleni segment LED RGB diode. LED diode podržavaju maksimalnu struju do 40mA. Struja RGB diode je ograničena putem otpornika s vrijednosti otpora:

$$R = \frac{U - U_L}{I_L}, \quad (4.1)$$

gdje je  $U$  napon napajanja (5V),  $U_L$  je pad napona na LED diodi, dok je  $I_L$  struja kroz LED diodu. Crvena LED dioda ima pad napona vrijednosti od 1,8 V, a zelena pad napona od 2,8 V. Uvrštavanjem u izraz (4.1) uz  $U = 5$  V, za željene vrijednosti struje približne vrijednosti 20mA dobivene se vrijednosti otpornika 150Ω za crvenu, te 100Ω za zelenu LED diodu. Za promjene stanja zaključanosti Arduino raspolaže informacijama koje daju senzor otiska prsta, RFID senzor, te podacima koje mu šalje Arduino Nano na kojeg je spojena numerička tipkovnica. Komunikacija između razvojnih pločica vrši se serijskim putem, tako da je Tx pin Arduino Nano spojen na Rx pin Arduino Uno razvojne pločice. Arduino Uno zvučnu signalizaciju ostvaruje korištenjem zujalica spojenih na D6 i D7. Shema spajanja razvojne pločice Arduino Uno prikazana je na slici 4.4.

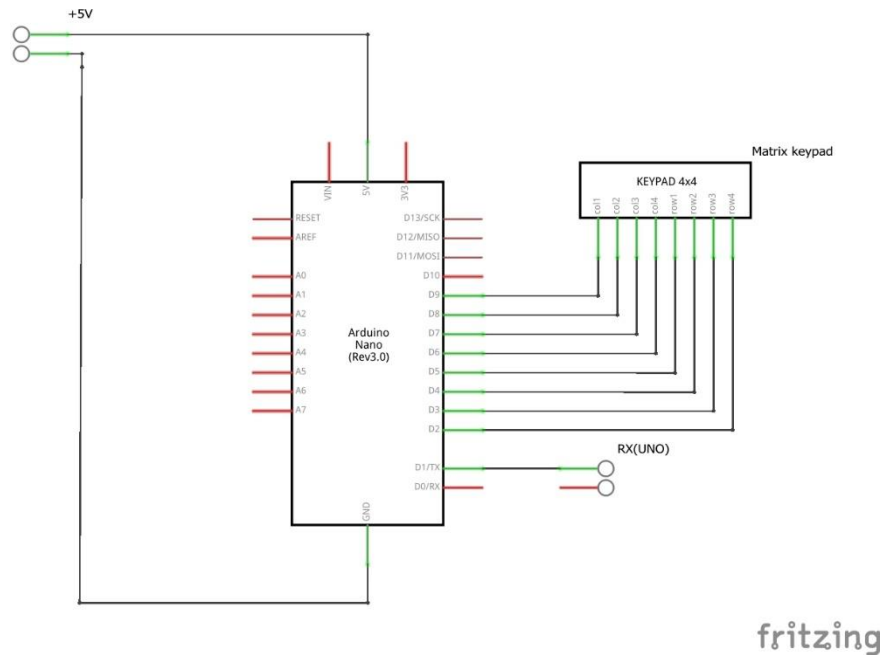


Slika 4.4. Shema spoja razvojne pločice Arduino Uno



#### 4.4 Shema spoja razvojne pločice Arduino Nano

Pošto numerička tipkovnica zahtjeva veliki broj pinova (njih 8) Arduino Nano je korišten isključivo za potrebe tipkovnice. Ova razvojna pločica očitava pritisak tipke na tipkovnici te putem programske podrške, pretvara taj podatak u određeni znak koji odgovara pritisnutoj tipki. Taj znak se putem serijske komunikacije, šalje na Arduino Uno. Shema spajanja razvojne pločice Arduino Nano prikazana je na slici 4.6.



Slika 4.6. Shema spoja razvojne pločice Arduino Nano

## 5. PROGRAMSKA PODRŠKA

Zbog višestrukog korištenja razvojnih pločica te učestalosti ponavljanja pojedinih dijelova koda u nastavku je prikazan program razvojne pločice Arduino Uno.

U početku koda potrebno je deklarirati pinove na koje su spojene komponente, što je ostvareno preko integer varijabli.

```
//deklariranje pinova  
const int R=5,G=4,zuja=6,sirena=7;
```

Kako bi koristili funkciju isDigit potrebno je uključiti biblioteku ctype. Funkcija isDigit vraća stanje logičke istine ukoliko je ulazni char argument znamenka. Nadalje, definirane su varijable u kojima je spremljena lozinka, broj pokušaja, te stanje ukoliko su svi pokušaji unošenja lozinke na tipkovnici iskorišteni i ukoliko je isteklo vrijeme potrebno za isključivanje alarmnog stanja. U nastavku su uključene biblioteke za senzor otiska prsta i RFID senzora.

```
//_____Tipkovnica_____
```

```
#include <ctype.h> //isDigit naredba
```

```
const String lozinka="2023";  
String unos=""; //spremanje unos lozinke
```

```
int pokusaji=0;
```

```
bool np=0; // neuspjeli pokušaji
```

```
//_____RFID_____
```

```
#include <SPI.h>  
#include <MFRC522.h>
```

```
#define SS_PIN 10  
#define RST_PIN 9  
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Pošto se serijska komunikacija koristi za komunikaciju između tri razvojne pločice, te senzora otiska prstiju, a Arduino Uno posjeduje samo jedan hardware-ski serijski port, potrebno je definirati dodatni software-ski serijski port pomoću funkcije SoftwareSerial.

```
//_____Fingerprint_____
#include <Adafruit_Fingerprint.h>

#if (defined(__AVR__) || defined(ESP8266)) && !defined(__AVR_ATMega2560__)
SoftwareSerial mySerial(2, 3);
#else
#define mySerial Serial1
#endif
```

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

```
int otisak=-1;
```

Da bi ostvarili točnije mjerenje vremena u milisekundama, korištena je funkcija millis() koja vraća vrijeme rada u milisekundama. Pomoću te funkcije ostvaren je brojač od 30 sekundi, koje je predviđeno da korisnik stigne zatvoriti i zaključati vrata nakon uključivanja zaključanog stanja. Za ovaj brojač potrebno je definirati interval utišanog djela zvučnog signala (u milisekundama) te varijablu koja će spremati koliko se puta taj interval ponovio. U ovom dijelu je također definirana i varijabla stanja otvorenosti vrata kao i varijabla stanja zaključanosti odnosno aktiviranog alarmnog stanja, te varijabla prethodnog stanja kako bi se kasnije moglo uočiti da li je došlo do promijene stanja.

```
//_____Vrijeme unosa_____
#include <millisDelay.h>
const unsigned long tu = 30000; //vrijeme čekanja unosa
millisDelay timer;

bool otvoreno;
bool zakljucano=false,pzakljucano=true; // status zakljucavanja,prethodni status
bool pt=0; //pritisnuta tipka

bool pz=false; //početak zaključavanja
int pzc=0; //početak zaključavanja counter sekunda
```

```
unsigned long previousMillis = 0;
const long interval = 990;
```

Funkcija setup() se izvršava samo prilikom uključene ili resetiranja razvojne pločice. Unutar nje inicijalizirani su korišteni komunikacijski protokoli, te su definirani digitalni pinovi koji su korišteni kao ulazi ili izlazi.

```
void setup() {

    Serial.begin(9600);
    while (!Serial);
    delay(100);

    SPI.begin();
    mfr522.PCD_Init();

    finger.begin(57600);
    delay(5);
    if (finger.verifyPassword()) {

    } else {

        while (1) { delay(1); }
    }

    pinMode(R,OUTPUT); //crvena LED
    pinMode(G,OUTPUT); //zelena LED
    pinMode(zuja,OUTPUT); //zelena LED
    pinMode(sirena,OUTPUT);
    digitalWrite(G, 1);

}
```

Funkcija loop() je beskonačna petlja, u kojoj je napisan glavni program. Kako bi se pokazao status alarmnog stanja, na početku funkcije se uključuje, odnosno isključuje crvena i zelena LED dioda prema bool varijabli postavljenoj za praćenje tog stanja.

```
digitalWrite(R, zakljucano);
digitalWrite(G, !zakljucano);
```

U integer varijablu otisak sprema se rezultat funkcije `getFingerprintIDez()`. Ukoliko se dobivena slika otiska podudara sa spremljenom slikom otiska prsta, tada rezultat integer varijable postaje jednak ID-u otiska spremljenog na senzoru. Ukoliko ne dođe do podudaranja otiska, integer varijabla poprimi vrijednost -1. U slučaju podudaranja mijenja se alarmno stanje, što podrazumijeva brisanje prethodnog unosa tipkovnice, te prikaz odgovarajućeg stanja na LED diodama, te ukoliko je novi status postao otključano, tada se izvršava funkcija za otključavanje. Također putem serijske komunikacije šalje se informacija na razvojnu pločicu Arduino Mega, kako bi se moglo započeti zapisivanje u memoriju o promjeni stanja te o kojem je statusu riječ, kao i metodi. Veliko slovo O označava da je došlo do otključavanja putem otiska prsta, dok malo slovo znači da je došlo do zaključavanja. Ukoliko vrijeme unosa lozinke istekne ili se dosegne maksimalan broj pokušaja na tipkovnici, alarmno stanje može se isključiti samo pomoću otiska prsta. Kada dođe to tog stanja varijabla np poprimi stanje logičke jedinice, te se uključi sirena. Kako je moguće isključiti alarmno stanje samo ovom metodom, potrebno je ove naredbe izvršavati u petlji dok se ne dobije odgovarajući otisak prsta. To je napravljeno pomoću do-while petlje. Ukoliko se stanje ovom metodom promijeni u otključano, tada se bool varijabla mijenja u logičku nulu te se petlja prekida, čime se također isključuje i sirena.

```
do{
//čekanje otiska prsta
otisak=getFingerprintIDez();

if(otisak!=-1){

    zakljucano=!zakljucano;
    unos="";
    digitalWrite(R, zakljucano);
    digitalWrite(G, !zakljucano);
    if(zakljucano==0){
        Otkljucavanje();
        Serial.print("O\n");
        np=0;
        digitalWrite(sirena, 0);
    }
    else if(zakljucano==1){
        Serial.print("o\n");
    }
    delay(3000);
}}while(np==1);
```



Ukoliko su vrata otvorena, na analognom pinu Arduino Uno razvojne pločice dolazi stanje visokog napona. Pošto očitavamo analognu vrijednost koja se sprema u digitalno stanje, potrebno je napraviti uvjet kako bi se prepoznalo o kojoj se logičkoj razini radi. Arduino može prepoznati maksimalno 1024 različite vrijednosti napona, pa možemo reći da je visoka razina napona veća od četvrtine maksimalne izmjerene vrijednosti odnosno od 255 što je u vrijednosti napona od 1,25V. Ukoliko je izmjerena vrijednost manja od ovoga uvjeta, možemo reći da smo dobili vrijednost logičke nule te da su vrata zatvorena.

```
//__Status otvorenosti vrata_____
if (analogRead(A0)>=255){
    otvoreno=1;

}
else{
    otvoreno=0;
}
```

Kada se status alarmnog stanja postavi u status zaključano brojač počne brojati vrijeme u trajanju 30 sekundi prije uključivanja alarmnog stanja kako bi korisnik imao dovoljno vremena zatvoriti te zaključati vrata. Kao signalizacija o prolasku vremena, uključuje se zujalica kao zvučna signalizacija u trajanju od 10ms u intervalu od jedne sekunde što znači da je 990ms zujalica utišana. Ponavljanjem intervala, uvećava se brojač koji broji koliko se interval puta ponovio odnosno koliko je sekundi prošlo. Nakon 30 sekundi uključuje se alarmno stanje. Kada se u alarmnom stanju otvore vrata započinje novi brojač od 30 sekundi koji služi kako bi korisnik stigao isključiti alarmno stanje (unošenjem lozinke, RFID karticom ili otiskom prsta) u zadanom vremenu. Ukoliko to ne uspije, izvršava se funkcija kojom se uključuje stanje neuspjelog isključivanja alarmnog stanja te je to stanje moguće isključiti samo otiskom prsta.

```
//_____
if(zakljucano){
    if(!pz){
        unsigned long currentMillis = millis();
        if (currentMillis - previousMillis >= interval) {
            previousMillis = currentMillis;
            digitalWrite(zuja, 1);
            delay(10);
            digitalWrite(zuja, 0);
            ++pzc;
            //vrijeme čekanja zatvaranja vrata
```

```

    if(pzc>=30){
        pz=1;
    }
}
}
else{
    if (otvoreno){
        if (!timer.isRunning()){
            timer.start(tu);
        }
        if(!pt){
            digitalWrite(zuja, 1);
        }
    }
}
}
//-----
if (timer.justFinished()){
    if(zakljucano) {NeuspjeloIskljucivanje();}
}
//-----

```

Kao i kod senzora otiska prsta potrebno je pokrenuti funkciju koja će započeti čekanje te očitavanje nove kartice. Ako novo prislonjena RFID kartica ima ID koji se podudara sa spremljenim IDom mijenja se status zaključanosti. Kao i kod senzora otiska prsta, promjenom alarmnog stanja, šalje se informacija na Arduino Mega te se izvršava skup naredbi za otključavanje odnosno zaključavanje.

```

//_____RFID čekanje kartice_____

if ( ! mfrc522.PICC_IsNewCardPresent())
{
    return;
}

if ( ! mfrc522.PICC_ReadCardSerial())
{
    return;
}

String kartica= "";
for (byte i = 0; i < mfrc522.uid.size; i++)
{

    kartica.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
}

```

```

    kartica.concat(String(mfrc522.uid.uidByte[i], HEX));
}

kartica.toUpperCase();

//provjera kartice
if (kartica.substring(1) == "B9 81 7C 73")
{
    zakljucano=!zakljucano;
    unos="";
    digitalWrite(R, zakljucano);
    digitalWrite(G, !zakljucano);
    if(zakljucano==0){
        Otkljucavanje();
        Serial.print("K\n");

    }
    else if(zakljucano==1){
        Serial.print("k\n");
    }

    delay(3000);
}

else {
    delay(3000);
}
}

```

Funkcija SerialEvent() se izvršava prilikom dolaska novih informacija putem serijske komunikacije.

```

void serialEvent() {
    while (Serial.available()) {

```

Zbog jednostavnijeg korištenja dolazne informacije iz serijske komunikacije, potrebno je informaciju spremi u novu varijblu.

```

    char t = (char)Serial.read(); // t -> tipka

```

Kada se vrata otvore u alarmnom stanju uključuje se zujalica. Kako nema potrebe da ona radi tokom unosa lozinke putem tipkovnice, postavlja se vrijednost pt=1, a zujalica ostaje isključena nakon izvršavanja naredbi koje uključe zujalicu u vremenu od 50ms te je isključe kako bi se dobila zvučna signalizacija pritiska tipke.

```

if (zakljucano) {pt=1;}

digitalWrite(zuja,1);
delay(50);
digitalWrite(zuja,0);

```

Dolazni znakovi iz serijske komunikacije se spremaju u string varijablu gdje se postojećoj vrijednosti dodaje novi dolazni znak. Kako se lozinka sastoji od 4 znaka odnosno broja, u nju se neće upisivati dodatni brojevi nakon upisa zadnjeg, četvrtog broja. Funkcija `isDigit()` osigurava da se u varijablu spremaju samo brojevi.

```

if (unos.length()<=3){
  if(isDigit(t)){
    unos+=t;
  }
}

```

Tipka (tj. znak) `#` označava potvrdu unosa. Ukoliko je korisnik unio ispravnu lozinku, tada se mijenja status alarmnog stanja te se varijabla `unos` briše, kako bi bila spremna za idući unos lozinke. Kao i kod senzora otiska prste te RFID senzora, ovisno o novom stanju, šalje se informacija o promijeni alarmnog stanja te se izvršavaju naredbe za novo alarmno stanje. Ukoliko je korisnik unio pogrešnu lozinku, integer varijabla, koja broji broj pokušaja, se uvećava za 1 te se također briše vrijednost varijable `unos` kako bi ona bila spremna za idući unos. Ukoliko korisnik iskoristi 3 pokušaja, pokreće se funkcija koja izvršava naredbe za neuspjelo isključivanje alarmnog stanja.

```

else if (t=='#'){
  if (unos==lozinka){
    zakljucano=!zakljucano;
    unos="";
    if(zakljucano==0){

      Otkljucavanje();
      Serial.print("P\n");
    }
    else if(zakljucano==1){
      Serial.print("p\n");
    }
  }
  else{
    if(zakljucano){
      ++pokusaji;
    }      unos="";
  }
}

```

```

if (pokusaji>=3){
    NeuspjeloIskljucivanje();
}

```

Tipka (tj. znak) \* služi za brisanje varijable unos. Ovo omogućuje ponovni upis lozinke, ukoliko je korisnik prilikom unosa napravio pogrešku.

```

if(t=='*'){
    unos="";
}

```

Funkcija Otkljucavanje() se koristi kada korisnik uspješno promijeni alarmno stanje iz zaključano u otključano. Ova funkcija postavlja varijable na početno stanje prije zaključavanja te završava brojanje brojača i isključuje rad zujalice.

```

void Otkljucavanje(){
    timer.finish();
    digitalWrite(zuja,0);
    pokusaji=0;
    pt=0;
    pz=0;
    pzc=0;
    unos="";
}

```

Funkcija NeuspjeloIskljucivanje() se izvodi ukoliko je korisnik unio pogrešnu lozinku više od 3 puta, ili ukoliko je isteklo vrijeme za isključivanje alarmnog stanja. Ova funkcija uključuje zujalicu i sirenu te postavlja vrijednost bool varijable na logičku jedinicu koja označava neuspjelo isključivanje alarmnog stanja.

```

void NeuspjeloIskljucivanje(){
    np=1;
    digitalWrite(zuja, 1);
    digitalWrite(sirena, 1);
}

```

Funkcija getFingerprintIDez() je funkcija koja izvršava naredbe za skeniranje novog otiska prsta putem senzora otiska prste te vraća integer vrijednost ID spremljenog otiska.

```

int getFingerprintIDez() {
    uint8_t p = finger.getImage();
}

```

```
if (p != FINGERPRINT_OK) return -1;

p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;

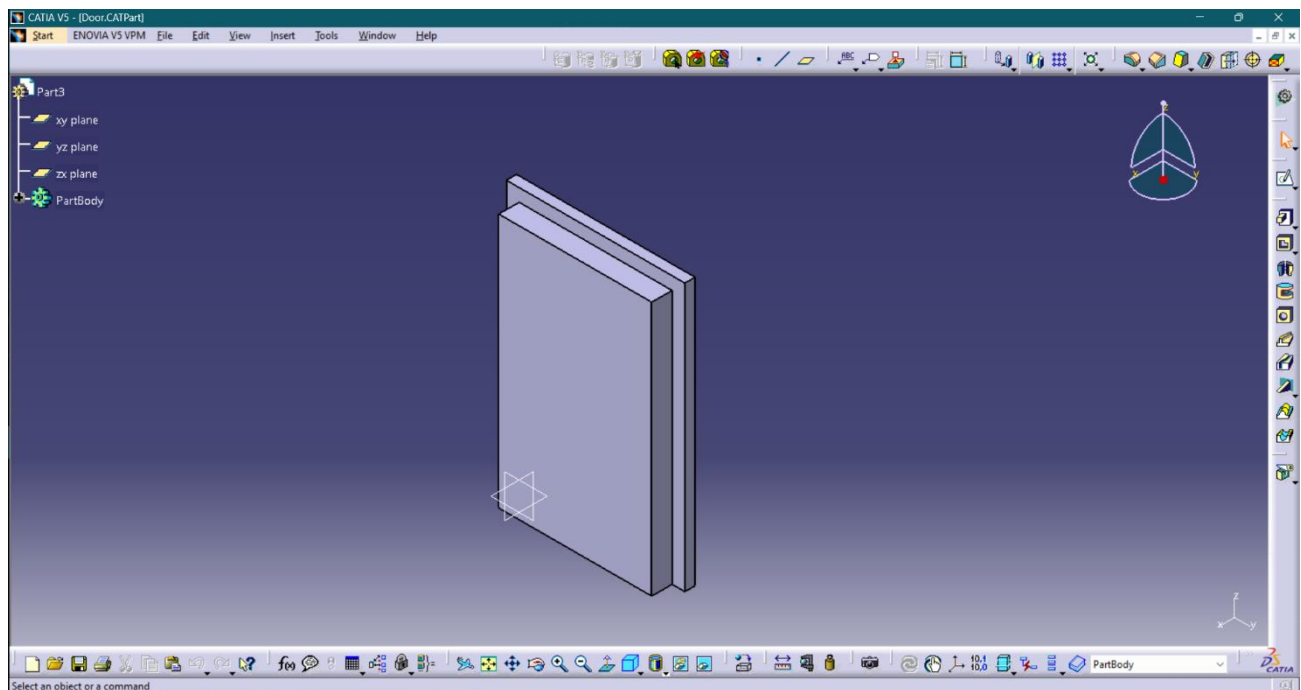
p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;

return finger.fingerID;
}
```

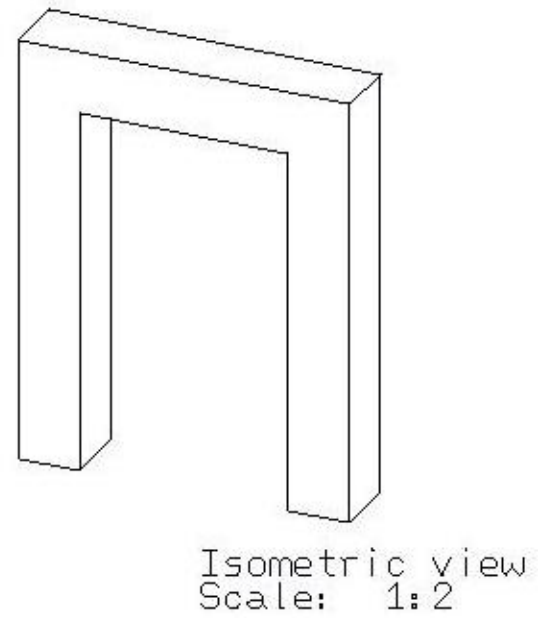
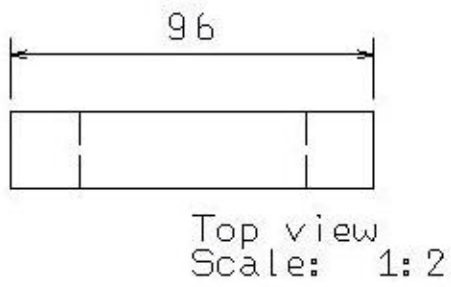
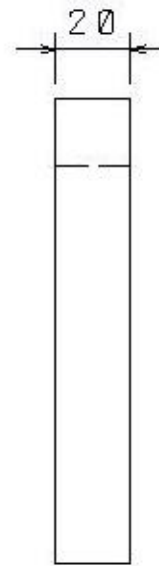
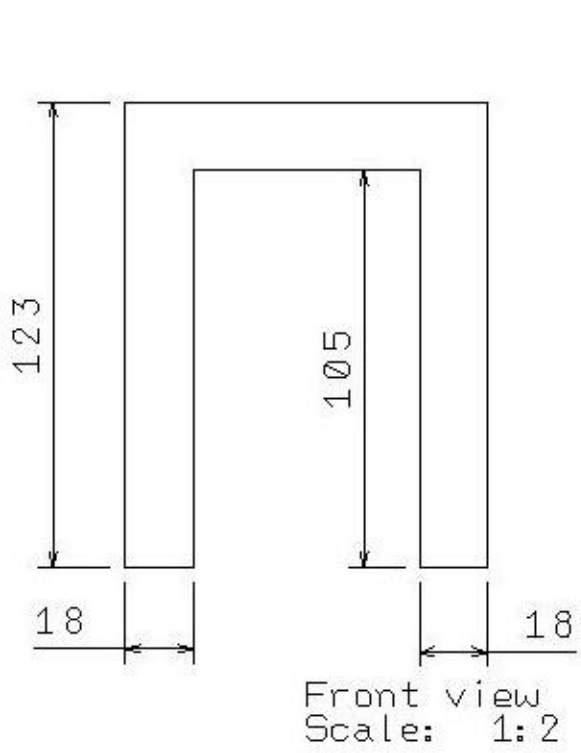
## 6. 3D PRINTANJE TE IZRADA MODELA U PROGRAMU CATIA

### 6.1. CATIA

Kao što je prethodno rečeno, potrebno je postaviti optokapler na okvir vrata, na način da je svjetlosni snop prekinut, kada su vrata zatvorena. U tu svrhu, izrađen je 3D model vrata u programskom paketu CATIA, koji je kasnije izrađen pomoću 3D printera. Catia je računalni program za izrađivanje 3D modela. Modeli se izrađuju tako što se odabere ravnina crtanja (nacrt, tlocrt, bokocrt) te se iz nacrtanog 2D prikaza izvuče treća dimenzija. Ovaj program nudi veliki izbor mogućnosti za detaljnu izradu modela te se stoga često koristi i u profesionalne svrhe. Catia ima mogućnost izvlačenja nacrt modela (nacrt, tlocrt, bokocrt), gdje se unutar programa može i izvući kote modela, kako bi se model moglo isprintati ili dokumentirati. Catia je pogodan alat za izradu modela koji se koriste za 3D printanje jer se modeli mogu spremirati u STL format. Izgled programa Catia prikazan je na slici 6.1. Na slikama 6.2., 6.3. i 6.4. su prikazani nacrt, tlocrt, bokocrt te izometrijski prikaz modela vrata, okvira vrata i modela za prekid svjetlosnog snopa optokaplera.

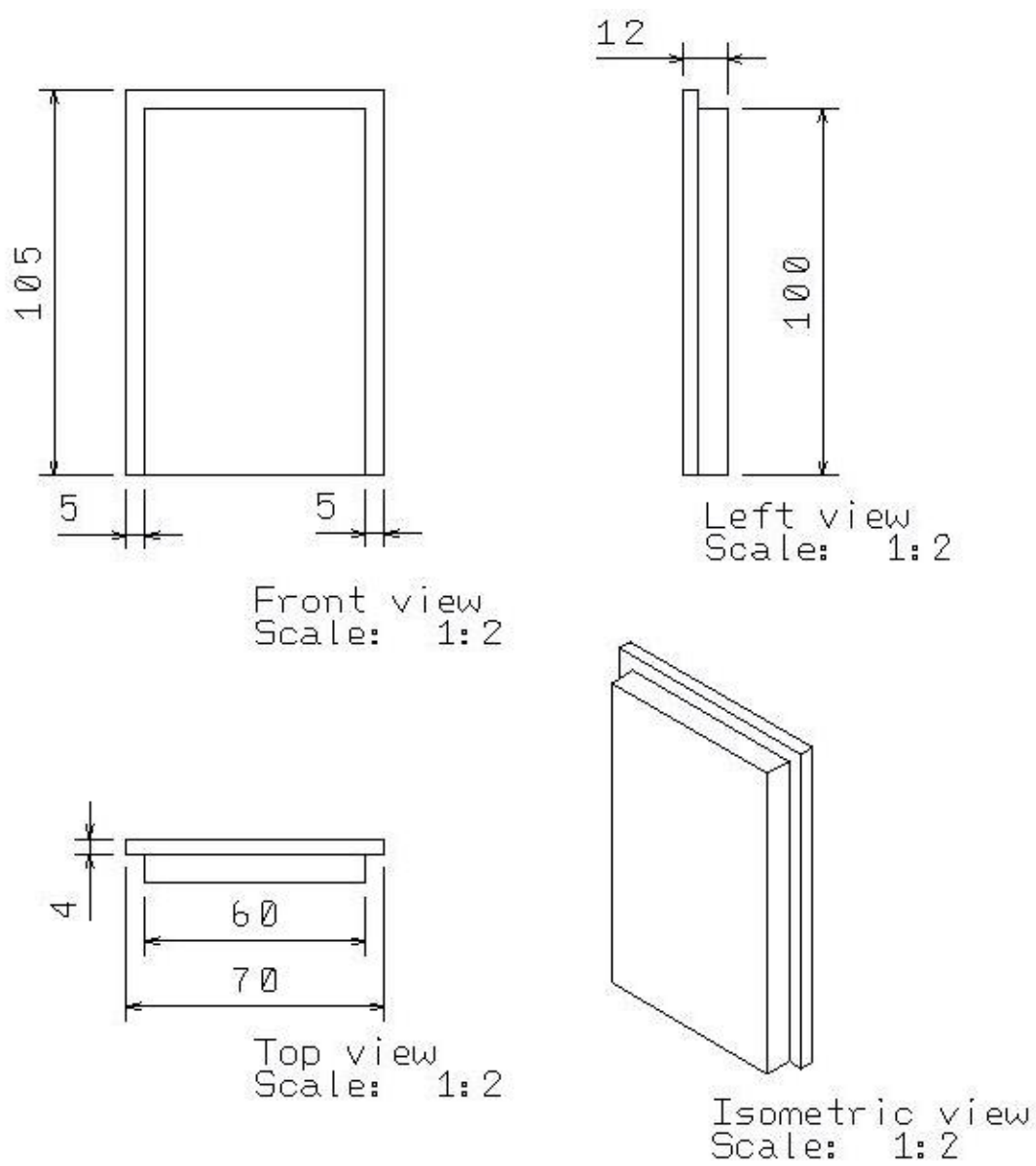


*Slika 6.1. Izgled programa CATIA*

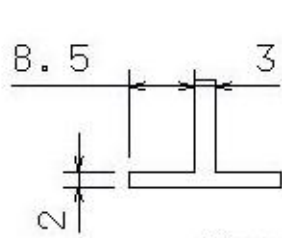


Slika 6.2. Nacrt modela okvira vrata

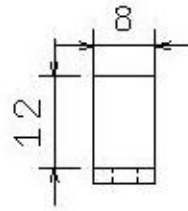




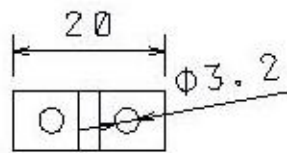
*Slika 6.3. Nacrt modela vrata*



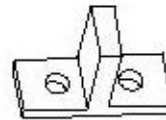
Front view  
Scale: 1:1



Left view  
Scale: 1:1



Top view  
Scale: 1:1

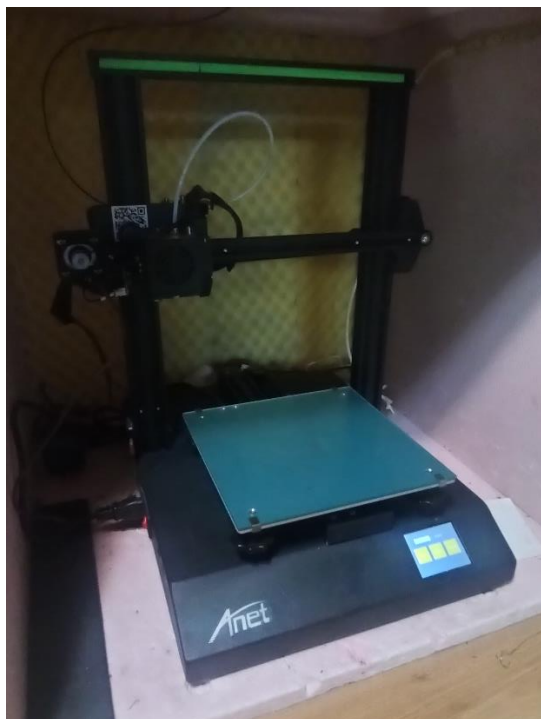


Isometric view  
Scale: 1:1

*Slika 6.4. Nacrt modela za prekid snopa svijetla optokaplera*

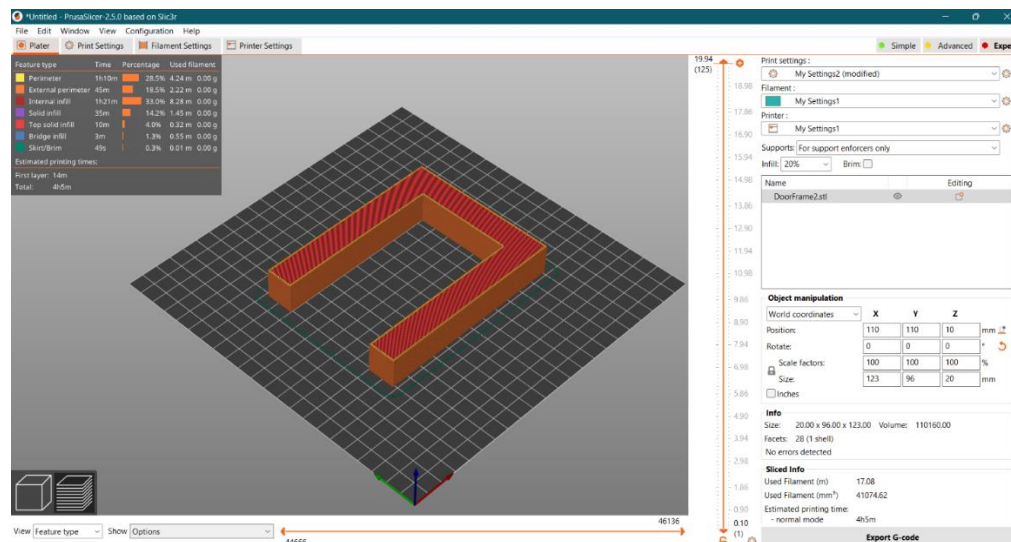
## 6.2. 3D printanje

3D printeri su uređaji koji računalne 3D modele izrađuju u 3D prostoru korištenjem plastike tako što je tope na određenoj temperaturi, ovisno o tipu plastike. Postoje više tipova plastike kao što su PLA, ABS, PETG, itd. Plastična nit (eng. *filament*) se pomoću koračnog motora spojenog s mehanizmom zupčanika, uvlači u ispisnu glavu (eng. *extruder*) u kojoj je ugrađen grijač. Zagrijavanjem plastične niti, ona se topi te se kroz glavu nanosi na podlogu (eng. *bed*). Podloga je zagrijana kako ne bi došlo do prebrzog hlađenja plastike te kako se prilikom printanja model ne bi odvojio od podloge. Za izradu modela korišten je 3D printer Anet ET4 pro, prikazan na slici 6.5.



*Slika 6.5. 3D printer Anet ET4 pro*

Kako bi model bio prepoznatljiv 3D printeru potrebno ga je pretvoriti u G-kod. Ovaj kod osim za upravljanje 3D printerom se koristi i za upravljanje CNC strojevima. Kako bi se pretvorilo model iz STL datoteke u G kod potrebno je koristiti računalni program slicer. Izgled programa slicer prikazan je na slici 6.6.



Slika 6.6. Izgled programa slicer (proizvođača Prusa)

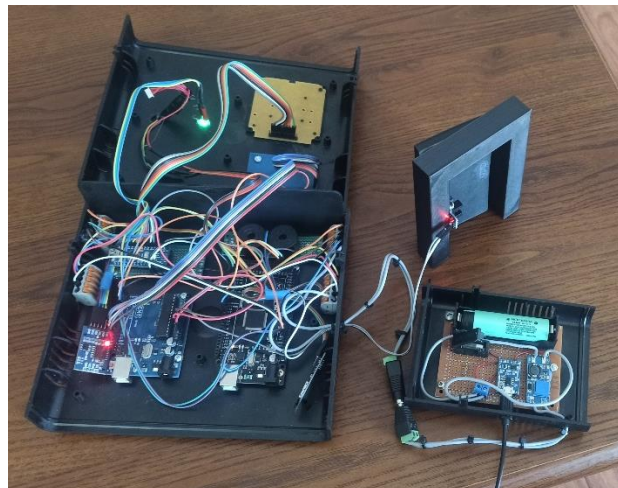
Unutar programa slicer potrebno je podesiti parametre poput temperature i ispunje printanja. Za printanje modela nije potrebno da je unutrašnjost 100% ispunjena plastikom, te bi takvo printanje predugo trajalo. Najčešće se koristi 20% do 35% ispunjenosti modela, a u ovom završnom radu korišteno je 20% ispunje. Također je potrebno podesiti temperature ekstrudera i podloge. S obzirom da je korištena PLA plastika, temperatura glave je postavljena na 210°C, a podloge na 60°C.

## 7. KONAČAN IZGLED RADA SIGURNOSNOG SUSTAVA

Nakon izrade rada na eksperimentalnoj pločici, te obavljenih testova rada sigurnosnog sustava, slijedilo je ubacivanje rada u kutiju zbog jednostavnog upravljanja sustava te stabilnosti kontakata elemenata. Na kutijama su izbušeni otvori za elemente i žice. Za krug napajanja komponente kruga su zalemljene na prototipnu pločicu kako bi se ostvario trajni spoj između komponenata. Nakon postavljanja te spajanja svih elemenata slijedilo je ponovno ispitivanje rada sigurnosnog sustava s čime je došlo do završetka izrade rada. Konačan izgled rada prikazan je na slikama 7.1. i 7.2.



*Slika 7.1. Konačan izgled rada sigurnosnog sustava*



*Slika 7.2. Konačan izgled rada sigurnosnog sustava (izgled unutar kutija)*

## **8. ZAKLJUČAK**

Protu-provalni alarmi znatno smanjuju rizik od provala jer zvučnom, svjetlosnom ili nekom drugom signalizacijom upozoravaju na provalu te se mogu poduzeti određene sigurnosne mjere. Uz prisutnost protu-provalnog alarma, provalnicima se povećava rizik da ih se uhvati na djelu te će više izbjegavati sigurnosne sustave. Potvrda otiska prsta je jedna od najsigurnijih načina deaktiviranja alarmnog stanja jer je otisak prsta jako teško kopirati ukoliko provalnik nije došao do kontakta s korisnikom sustava. Razvojne pločice Arduino, zbog svoje nestabilnosti pri duljem i neprekinutom radu rada te manjku sigurnosnih protokola unutar same pločice, nije poželjno koristiti u profesionalne svrhe već se mogu koristiti za izradu modela sustava protu-provalnog alarma.

## 9. LITERATURA

- [1] s Interneta, <https://en.wikipedia.org/wiki/Arduino>
- [2] s Interneta, <http://art364.pbworks.com/w/page/13719084/Standalone%20Arduino>
- [3] s Interneta, <https://randomnerdtutorials.com/25-Arduino-shields/>
- [4] s Interneta, <https://soldered.com/hr/learn/kkm-senzor-otiska-prsta/>
- [5] s Interneta, <https://randomnerdtutorials.com/fingerprint-sensor-module-with-Arduino/>
- [6] s Interneta, <https://peppe8o.com/rfid-with-Arduino-uno-rc522-wiring-and-code/>
- [7] s Interneta, <https://techzeero.com/sensors-modules/ds3231-rtc-module/>
- [8] s Interneta, <https://components101.com/modules/micro-sd-card-module-pinout-features-datasheet-alternatives>
- [9] s Interneta, <https://www.circuits-diy.com/tp4056-battery-charger-module/>
- [10] s Interneta, <https://embedblog.eu/?p=712>

## 10. SAŽETAK I KLJUČNE RIJEČI

U sklopu ovog završnog rada izrađen je protu-provalni sustav koji ima mogućnost uključivanja, te isključivanja na tri načina: unošenjem lozinke na tipkovnici, RFID karticom te otiskom prsta. U tu svrhu svi senzori, tipkovnica i elementi zvučne signalizacije spojeni su na Arduino razvojne pločice za kojih je napisana odgovarajuća programska podrška. Zbog manjka pinova na Arduinu te potrebe za većom stabilnosti Arduino razvojne pločice, korišteno je više razvojnih pločica koje međusobno komuniciraju. Arduino Nano je korišten isključivo za potrebe tipkovnice, dok se Arduino Mega koristi isključivo za zapis podataka na mikro SD karticu. Arduino Uno ima ulogu centralne jedinice na koju su spojeni svi senzori kao i zvučni signalni elementi te služi za obradu svih podataka. Osim programske podrške izrađen je krug neprekinutog napajanja, ukoliko nestane napajanje iz gradske mreže. Naposljetku, cijeli sklop je montiran u kućište, te je izrađen model vrata putem 3D printera, kako bi se lakše demonstrirao rad sigurnosnog sustava odnosno protu-provalnog alarma.

Ključne riječi: Arduino, sigurnosni sustav, alarmno stanje, senzori



## **11. ABSTRACT AND KEY WORDS**

In this bachelors thesis a model of anti-burglar system was designed, being capable of arming and disarming in three ways: by entering a password on the keyboard, using an RFID card, and a fingerprint. In order to do this, it was necessary to connect sensors, a keyboard, and sound signaling elements to the Arduino development boards and write the appropriate program support. Due to the lack of pins on the Arduino and a need for the greater stability of the Arduino development board, multiple boards have been used that communicate with each other. The Arduino Nano has been used in order to handle numerical keyboard, while the Arduino Mega in order to write data to the micro SD card. Arduino Uno has the role of central unit, on which all the sensors and sound signaling elements are connected, and additionally it process all of the data. In addition to program support, it was also necessary to make a uninterruptable power supply in case of disturbances in the electrical grid. Finally, the entire system has been placed in the appropriate housing, and a door model was 3D printed, in order to demonstrate the workings of this security system.

Keywords: Arduino, security system, alarm state, sensors