

Automatska sigurnosna vrata

Herak, Marin

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:197616>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-06**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Marin Herak

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Mladen Crneković, dipl. ing.

Student:

Marin Herak

Zagreb, 2022.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se svojoj obitelji i prijateljima na potpori i razumijevanju, svim profesorima koji su mi svojim trudom i entuzijazmom prenijeli svoje znanje što je veoma pripomoglo pri izradi ovog završnog rada te svom mentoru prof. dr.sc. Mladenu Crnekoviću koji mi je dao ideju za sam rad i pomogao savjetima u savladavanju prepreka za ostvarivanje iste u praksi.

Marin Herak



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomatske ispite
Povjerenstvo za završne i diplomatske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 22 – 6 / 1	
Ur.broj: 15 - 1703 - 22 -	

ZAVRŠNI ZADATAK

Student: **Marin Herak** JMBAG: **0035220795**

Naslov rada na hrvatskom jeziku: **Automatska sigurnosna vrata**

Naslov rada na engleskom jeziku: **Automatic Security Door**

Opis zadatka:

Automatska vrata sa sigurnosnim otključavanjem danas su vrlo česta ne samo u tvrtkama već i u privatnim stambenim objektima. Potrebno je konstruirati sigurnosni mehanizam vrata koji u sebi uključuje otključavanje preko više medija koji mogu raditi istovremeno. Također, sigurnosni uređaj kontrolira dodatno zaključavanje u slučaju detekcije pokušaja provala. Dodavanje i brisanje korisnika koji su ovlašteni otvoriti vrata mora biti "user friendly".

U radu je potrebno:

- konstruirati mehanizam sigurnosnih vrata,
- odabrati potrebne senzore, aktuatora i upravljački mikrokontroler,
- projektirati upravljački PCB i predložiti strukturu upravljačkog programa,
- predložiti sučelje prema korisniku i način programiranja,
- procijeniti vrijednost uređaja.

Potrebno je navesti korištenu literaturu, te eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2021.

Zadatak zadao:

Prof. dr. sc. Mladen Crneković

Datum predaje rada:

1. rok: 24. 2. 2022.
2. rok (izvanredni): 6. 7. 2022.
3. rok: 22. 9. 2022.

Predvideni datumi obrane:

1. rok: 28. 2. – 4. 3. 2022.
2. rok (izvanredni): 8. 7. 2022.
3. rok: 26. 9. – 30. 9. 2022.

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA	III
SAŽETAK	IV
SUMMARY	V
1. UVOD	1
1.1. Postojeća rješenja.....	1
1.1.1. Elektroničke brave serije secuENTRY, TSE Prime i Home.....	1
1.1.2. ECKey – Bluetooth kontrola pristupa	4
2. REALIZACIJA AUTOMATSKIH SIGURNOSNIH VRATA.....	7
2.1. Prijedlog rješenja.....	7
3. ODABIR KOMPONENATA	8
3.1. Arduino Mega 2560.....	8
3.2. Tipkovnica.....	13
3.3. RFID-RC522	14
3.4. Bluetooth HM-10.....	17
3.5. Senzor otiska prsta.....	19
3.6. Elektromagnetska brava za vrata	21
3.7. Relej.....	22
4. SPAJANJE I TESTIRANJE.....	24
4.2. Električna shema.....	24
4.3. Eksperiment.....	25
4.4. Izrada prototipne pločice	27
4.5. Alternativna izrada PCB pločice.....	29
5. PROGRAMIRANJE	33
5.1. Dijagram toka	33
5.2. Arduino IDE	33
5.3. Deklaracija varijabli	35
5.4. Setup.....	37
5.5. Petlja.....	37
6. ZAKLJUČAK	52
LITERATURA.....	53
PRILOZI	54

POPIS SLIKA

Slika 1. Elektronička brava secuENTRY[1]	3
Slika 2. Aplikacija KeyApp za pametni telefon[1]	4
Slika 3. Prikaz korištenja mobitela umjesto klasične tipkovnice[2]	5
Slika 4. ECKey[2]	5
Slika 5. Arduino Mega 2560 razvojna platforma	8
Slika 6. USB standardi	11
Slika 7. Shematski prikaz Arduino Mega 2560 razvojne platforme	12
Slika 8. Tipkovnica 4x4	13
Slika 9. RFID čitač	14
Slika 10. Prikaz rada čitača	15
Slika 11. Shematski prikaz RC522 čitača	16
Slika 12. Bluetooth modul HM-10.....	17
Slika 13. Shematski prikaz Bluetooth modula HM-10.....	18
Slika 14. Senzor otiska prsta	19
Slika 15. Princip rada senzora otiska prsta.....	20
Slika 16. Elektromagnetska brava	21
Slika 17. Relej	22
Slika 18. Shematski prikaz releja	23
Slika 19. Električna shema	24
Slika 20. Eksperimentalni spoj.....	25
Slika 21. Ikona aplikacije za pametni telefon	26
Slika 22. Sučelje aplikacije	26
Slika 23. Prototipna pločica	27
Slika 24. Model kućišta.....	28
Slika 25. Kućište	28
Slika 26. Top layer PCB-a.....	29
Slika 27. Bottom layer PCB-a	30
Slika 28. 3D prikaz.....	31
Slika 29. 3D prikaz odozgo	31
Slika 30. Gerber file top layer-a	32
Slika 31. Gerber file bottom layer-a.....	32
Slika 32. Dijagram toka.....	33
Slika 33. Arduino IDE sučelje.....	34

POPIS TABLICA

Tablica 1. Značajke elektroničke brave.....	2
Tablica 2. Specifikacije Arduino Mega razvojne platforme	9
Tablica 3. Specifikacije RC522 čitača	15
Tablica 4. Specifikacije Bluetooth modula HM-10	17
Tablica 5. Specifikacije senzora otiska prsta	20
Tablica 6. Specifikacije elektromagnetske brave	21
Tablica 7. Specifikacije releja	23

SAŽETAK

U sklopu završnog rada u početku je dan kratak uvod o značaju upotrebe otvaranja vrata i osvrt na slične tehnologije koje se danas primjenjuju. U drugom dijelu je dan popis i opis komponenata koja se koristila za realizaciju samog zadatka, a to je otvaranje vrata. Treći dio obuhvaća električni dio, od spajanja sheme pojedinih komponenti do povezivanja svih odabranih modula u cjelinu, te opis rada cijelog sklopa kao zasebnog uređaja. Nadalje, dana je i alternativna mogućnost izvedbe projekta pomoću jedne zasebne tiskane pločice. U četvrtom dijelu je opisan način i logika razmišljanja kod samog programiranja sklopa za izvršavanje zadane funkcije. A u zadnjem, petom dijelu, dan je zaključak nakon provedbe cijelog završnog rada, od teorijskog ka praktičnom dijelu.

SUMMARY

This undergraduate thesis provides a short introduction to the use of door opening and its significance with an overview of similar technologies applied today. The second part of the thesis describes the list and description of the equipment used for the execution of the assignment i.e. electronic lock opening. Furthermore, the third part includes the electronics- from connecting individual components to merging the selected modules into a functional unit, as well as describing the work of the electronic assembly as a single device. Furthermore, the thesis defines an alternate possibility to project implementation by using a separate P.C. board. The fourth part elaborates on the way of thinking and the logic behind the programming of the needed assembly. The last part of the thesis provides the conclusion, which arises from theory to practice.

1. UVOD

Automatska vrata sa sigurnosnim otključavanjem, u današnje vrijeme kada je čovjekov ritam užurban, znatno mogu olakšati pristup nekom objektu i sl. Vrata trebaju biti realizirana da se njima može upravljati na različite načine (tipkovnica, RFID, otisak prsta, pametni telefon) istovremeno. Ovaj rad sastavljen je pomoću mikroupravljačkih pločica i već gotove mobilne aplikacije. Današnjim razvojem tehnologije sve se više zamjenjuje klasični mehanički ključ jer je jednostavniji i brži ulazak u prostoriju.

1.1. Postojeća rješenja

Mogućnosti koje pruža današnje tržište uključuju raznorazna rješenja kod beskontaktnog otključavanja i zaključavanja brave. U nastavku su navedena neka od danas popularnih ponuda na tržištu.

1.1.1. Elektroničke brave serije secuENTRY, TSE Prime i Home

Njemačka tvrtka BURG-WÄCHTER čiji slogan glasi „Bez sigurnosti nema slobode“ proizvodi inovativne elektroničke brave koje zamjenjuju alternativne, odnosno klasične današnje brave. Tvrtka svakodnevno radi na tome da svijet postane malo sigurniji – i to već 100 godina. S radom je započela još 1920. godine, a u međuvremenu je postala poznata marka u području sigurnosti. Sa ovim sustavom elektroničkih brava možete otvoriti svoja vrata na jednostavan način sa otiskom prsta, lozinkom ili svojim mobitelom. Uz ove elektroničke brave izgubljeni ključevi su prošlost.

Set elektroničke brave s tipkovnicom secuENTRY easy 5601 Pincode nudi iznimnu fleksibilnost u području sustava otvaranja vrata bez ključa. Posebni sigurnosni cilindar omogućuje otvaranje vrata putem 6-znamenkastog pristupnog koda ili putem KeyApp aplikacije na pametnom telefonu. KeyApp aplikacija radi na radio vezi Bluetooth 4.0 LE (Low Energy) koji u usporedbi s klasičnim Bluetooth-om je namijenjen pružanju znatno smanjene potrošnje energije i troškova uz održavanje sličnog komunikacijskog dometa. Bluetooth Low Energy tehnologija radi u istom opsegu spektra (ISM opseg 2.400–2.4835 GHz) kao klasična Bluetooth tehnologija.

U elektroničku bravu može se programirati do 50 kodova preko osvijetljenih dodirnih pločica odnosno tipkovnice. Također moguće je 1 milijun različitih kombinacija kodova.

Tablica 1. Značajke elektroničke brave

Širina	59 mm
Visina	132 mm
Duljina	72 mm
Vanjska dužina	59 mm
Dužina unutarnja	59 mm
Vrsta brave	Elektronički profilni cilindar
Oprema	Osvijetljena tipkovnica, mehaničko zaključavanje u slučaju nužde
Baterije	6 kom
Napon baterije	1,5 V
Tip baterije	Mignon AA
Rukovanje	Kod za pristup, pametni telefon
Temperatura pogona	Od -15°C do +50°C
Područja primjene	Sobna vrata, ulazna vrata
Boja	Crno-srebrne boje
Prikladno za	Vrata debljine do 118 mm (59/59)
Preduvjeti sustava	Pametni telefon, aplikacija, iPhone 4s ili noviji, Android 4.4 ili noviji
Debljina vrata	0 mm – 118 mm
Dodatna svojstva	Bez bušenja, bez kabela
Tip cilindra	Cilindri secuENTRY, profilni cilindri TSE s radijalnom vezom
Prijenos podataka	Radijska veza

Vrsta zaštite	IP65
Težina (neto)	750 g



Slika 1. Elektronička brava secuENTRY[1]

Aplikacija KeyApp koja je prikazana na slici [Slika 2.] može se besplatno preuzeti u App store ili Google Play Store i jednostavno i lako se otvara cilindar pomoću pametnog telefona. Također se pomoću nje može iščitati i prikazati povijest.



Slika 2. Aplikacija KeyApp za pametni telefon[1]

Ovo je prikazana samo jedna od mnogo serija elektroničkih brava. Svaka se razlikuje ovisno o namjeni i mjestu gdje se koristi, tako je ovisno o tome moguća i nadogradnja sa otiskom prsta i daljinskim (ključ s radio vezom).

1.1.2. ECKey – Bluetooth kontrola pristupa

ECKey tehnologija se vodi sloganom „Pretvorite svoj mobitel u ključ...“. Ovakva tehnologija omogućava da pomoću Bluetooth veze odnosno mobilnim uređajima da služe kao sigurnosni ključ za otključavanje/zaključavanje automobila, ureda, kuća, stanova, garaža, dvorišnih vrata, parking barijera, podiznih stupova, hotela, sportskih klubova i alarmnih sustava. Jedan ključ otključava sve! Nova briljantna ECKey tehnologija jednostavno se integrira s postojećim vratima. Nakon jednokratne registracije, mobilni uređaj će automatski zaključati vrata na odlasku, i automatski otključati vrata prilikom povratka. Pri tome nije potrebno pritiskati nikakve gumbе, niti tražiti ključeve, čak ni izvaditi mobilni uređaj iz džepa ili torbe. ECKey uređaj ima mogućnost identifikacije mobitela na siguran način, koristeći pri tome jedinstvenu Bluetooth adresu i sigurnu 128-bitnu enkripciju. Koriste se dalekometni Proximity čitači dometa do 10 m za „hands-free“ kontrolu pristupa. Tvrtka koja nudi navedeno rješenje je UNIS Zagreb d.o.o. specijalizirana za projektiranje, proizvodnju i prodaju različitih vrsta proizvoda baziranih na sigurnosnom programu.

ECkey uređaji su proizvedeni u različitim modelima i s različitom primjenom: mogu biti samostalni ili integrirani s postojećim sustavima, mogu imati offline ili online kontrolu pristupa.

Jedna od mogućnosti koje nudi ECkey tehnologija jest unaprijed podešena virtualna tipkovnica koja se koristi umjesto klasične tipkovnice na zidu, i nalazi se na mobitelu koji se potom koristi za davanje pristupa. Može se instalirati s unutarnje strane zbog zaštite od nevremena i vandalizma.



Slika 3. Prikaz korištenja mobitela umjesto klasične tipkovnice[2]

ECKey je razvijen na bazi Bluetooth uređaja i instalira se u neposrednoj blizini elektro-brave (elektro-prihvatnika, garažnih ili dvorišnih vrata, alarmnog sustava, sustava za kontrolu pristupa, i sl.).



Slika 4. ECkey[2]

Prednosti ECKey tehnologije:

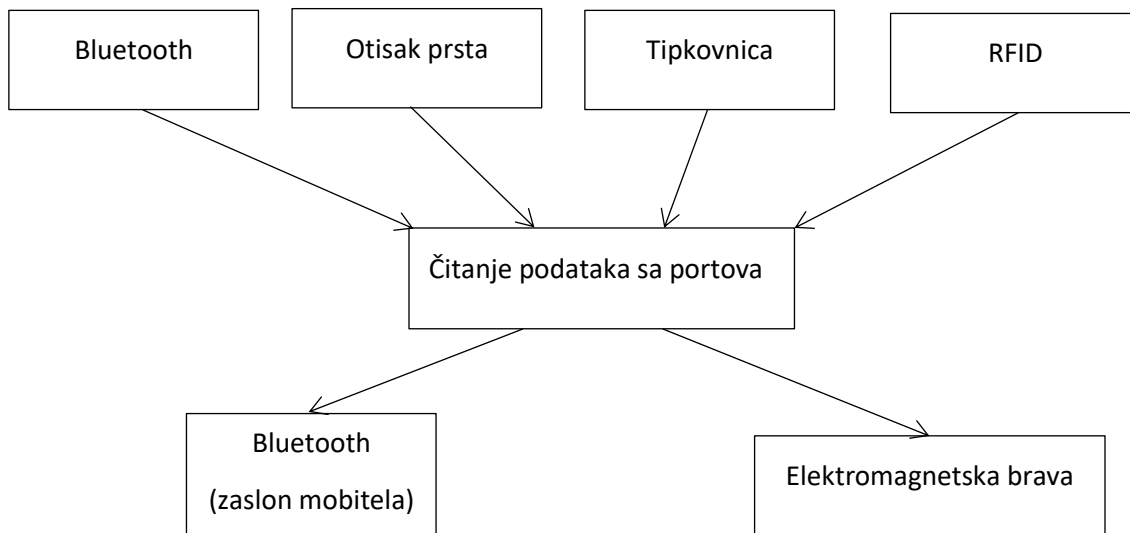
- Zaključavanje i otključavanje bez ruku – mobitel ostaje u džepu ili torbi!
- Jedan uređaj manje za nošenje - nema potrebe za nošenjem daljinskog upravljača ili magnetske kartice koja se može oštetiti nošenjem.
- Nema preturanja po džepovima ili torbi zbog traženja ključeva ili daljinskog upravljača.
- Može se koristiti svaki Bluetooth uređaj bez ikakve instalacije softvera.
- Brava i mobitel ostaju nevidljivi s 128 bit-a enkripcijom.
- Može se dodati na bilo koju elektro-bravu.
- Registracija i administriranje više uređaja.
- Radi sa svim Bluetooth uređajima (čak i s onima na kojima je Bluetooth parcijalno onemogućen)
- Uređaj ne mora biti vidljiv (može biti skriven za ostale korisnike u određenom opsegu).

2. REALIZACIJA AUTOMATSKIH SIGURNOSNIH VRATA

Cilj završnog rada je realizacija automatskih sigurnosnih vrata. Najprije je potrebno odabrati odgovarajuće komponente te ih povezati u funkcionalnu cjelinu.

2.1. Prijedlog rješenja

Za upravljanje radom koristiti će se Arduino mikroupravljač iz razloga što je cjenovno prihvatljiv, a ne zahtijeva visoku razinu znanja prilikom programiranja. Na prikazanom dijagramu toka prikazan je okvirni prijedlog rješenja.

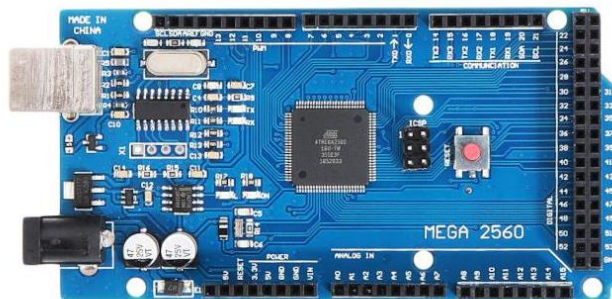


3. ODABIR KOMPONENATA

Komponente se odabiru prema njihovoj cjenovnoj isplativosti, a da pritom zadovoljavaju postavljene tehničke zahtjeve. Trebaju se lako spajati i trebaju imati biblioteke koje su lako dostupne.

3.1. *Arduino Mega 2560*

Arduino Mega prikazana na slici [Slika 5.] razvojna je platforma temeljena na ATmega2560 mikroupravljaču. Specifikacije Arduino Mega razvojne platforme prikazane su u tablici [Tablica 2.]. Ima 54 digitalna ulazno/izlazna pina, 16 analognih ulaza, UART (engl. universal asynchronous receiver/transmitter), 16 Mhz kristalni oscilator, tipku RESET, utičnicu za napajanje i USB priključak. Atmega2560 je upravljačka jedinica na kojem se nalaze ulazne, izlazne jedinice i programski kod koji se neprestano izvršava.



Slika 5. Arduino Mega 2560 razvojna platforma

Tablica 2. Specifikacije Arduino Mega razvojne platforme

Mikrokontroler	Atmega2560
Radni napon	5V
Ulazni napon (preporučeno)	7-12V
Ulazni napon (granica)	6-20V
Digitalni I/U pinovi	54 (od kojih su 15 PWM izlazi)
Analogni ulazni pinovi	16
DC struja po I/U pinovima	20 mA
DC struja za 3.3V pin	50 mA
Flash memorija	256 KB od kojih 8 KB koristi bootloader
SRAM	8 KB
EEPROM	4KB
Brzina rada	16 MHz
Ugrađena LED svjetla	13
Dužina	101.52 mm
Širina	53.3 mm
Masa	37 g

Mikroupravljač radi na naponskoj razini od 5V. Prilikom spajanja ulaznih elemenata treba paziti da naponska razina ne prijeđe radnu razinu kako ne bi došlo do oštećenja. Programiranje se vrši u Arduino IDE programskom sučelju, a program se učitava putem USB komunikacijskog porta. Arduino Mega 2560 je odabran jer ima dovoljno pinova i komunikacijskih sučelja kako bi se na njega mogle spojiti sve komponente, a ima veliki broj potrebnih biblioteka koje su mu na raspolaganju.

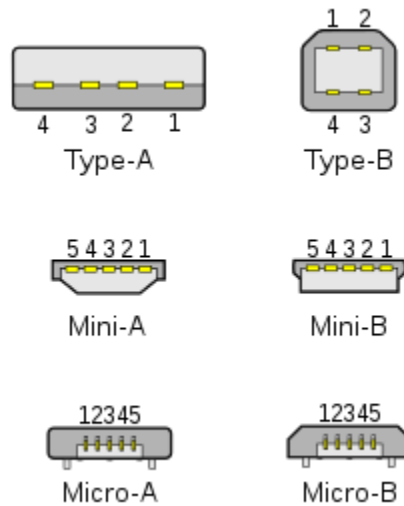
Arduino se može napajati izravno iz USB konektora ili preko priključka za napajanje. Napajanje se preko priključka za napajanje može postići preko baterije, ali i preko bilo kojeg pretvarača izmjenične u istosmjernu struju (npr. punjač za laptop), ali uz uvjet da je ulazni napon maksimalno 12 V zbog toga jer se na naponu većem od tog iznosa sklop za zaštitu od previsokog napona može pregrijati i uništiti cijelu pločicu.

Mikroprocesor ATmega 2560 je zapravo glavni dio Arduina jer upravlja njegovim radom, šalje signale na određene digitalne i analogne izlaze, provjerava da li je u sustavu sve u redu te izvršava kod koji je pohranjen u 256 KB memoriji. Svih 54 digitalnih pinova mogu biti kao izlazni ili ulazni pinovi, a 16 analognih samo kao ulazni. Arduino komunicira sa računalom preko USB-a (engl. Universal Serial Bus), serijskom vezom. Na Arduino se vrlo jednostavno mogu ugraditi dodatni moduli, koji proširuju njegove sposobnosti. Svaki modul ima zasebno napajanje za sebe, a kada se ugradi na Arduino, vanjski priključak napajanja Arduina treba se isključiti jer je Arduino konstruiran tako da ima samo jedno napajanje.

Univerzalna serijska sabirnica (engl. Universal Serial Bus, USB) je tehnološko rješenje za komunikaciju računala s vanjskim uređajima pri čemu se podatci razmjenjuju serijski relativno velikom brzinom. USB je zamijenio razna dotadašnja serijska i paralelna sučelja na računalima.

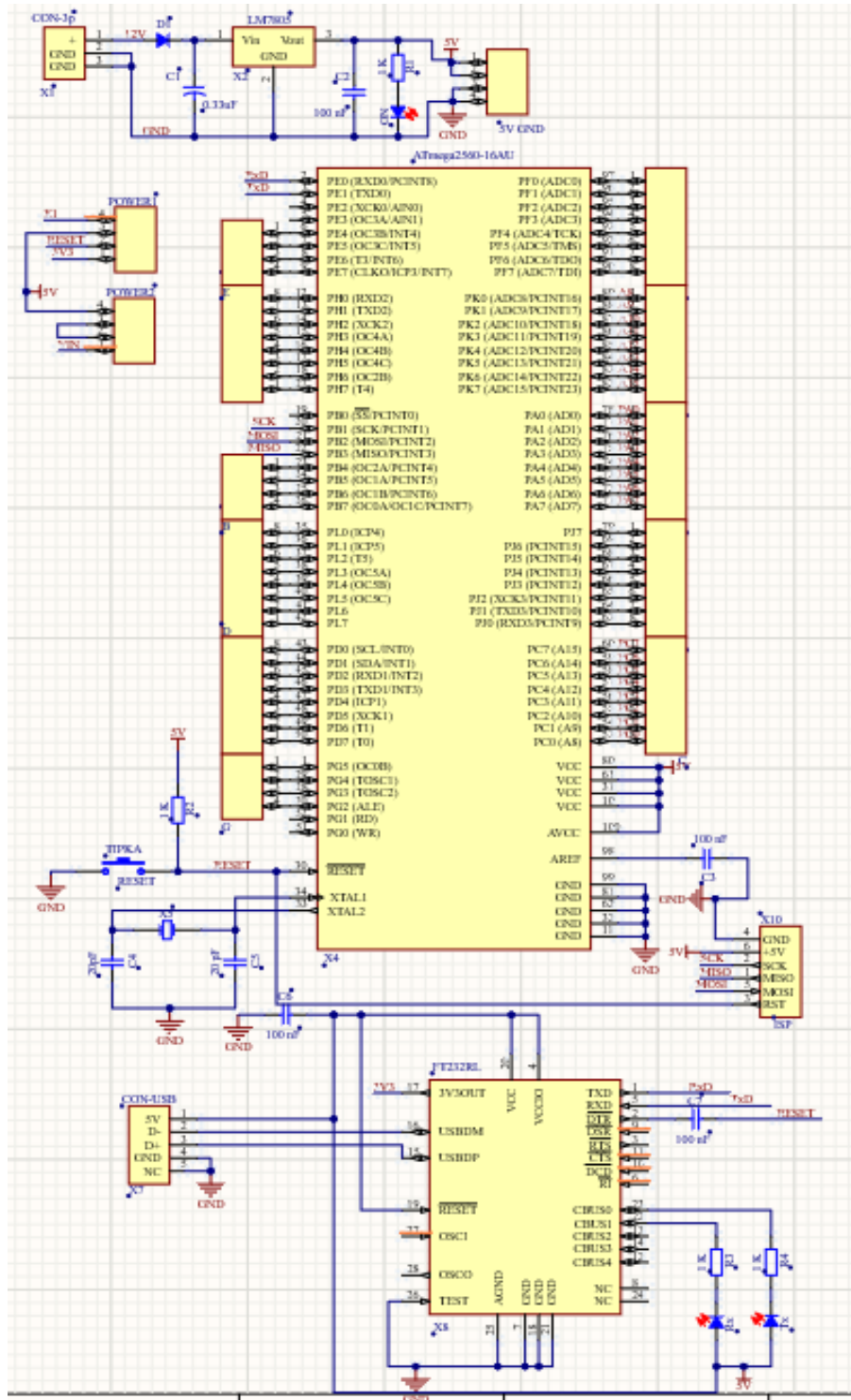
Cilj tehnologije USB jest rasterećivanje glavne sabirnice računala od posebnih kartica za proširenje, kao i olakšavanje umetanja i odvajanja vanjskih uređaja (plug-and-play) bez potrebe za ponovno pokretanje računala (hot swapping).

Postoji više standarda i tipova koji su prikazani na slici [Slika 6.], a konkretno kod Arduina Mega2560 se najčešće koristi tip B.



Slika 6. USB standardi

Na slici [Slika 7.] prikazan je shematski prikaz razvojne platforme zasebno nacrtan u programu Altium Designer-u u svrhu izrade zasebne tiskane pločice.



Slika 7. Shematski prikaz Arduino Mega 2560 razvojne platforme

3.2. Tipkovnica

Unos PIN-a vrši se putem tipkovnice 4x4 prikazanoj na slici [Slika 8.]. Odabrana je jer ima dovoljan broj tipki, a troši malo struje. Ova tipkovnica, koja se sastoji od 16 tipki, omogućuje interakciju korisnika i mikroupravljača. Podloga pruža jednostavan način za postavljanje tipkovnice na različita mjesta.



Slika 8. Tipkovnica 4x4

Najvažnije specifikacije:

- Maksimalna snaga: 24 VDC, 30 mA
- Sučelje: 8 pinski pristup 4x4 matrici
- Temperatura rada: (0 do 50 °C)
- Dimenzije: Tipkalo (6,9 x 7,6 cm), konektor (2,0 x 8,8 cm)

3.3. *RFID-RC522*

RFID-RC522 prikazan koristi se jer ima malu potrošnju, a odgovara tehničkim zahtjevima. Spaja se na komunikacijske pinove Arduino Mege 2560. Arduino Mega 2560 prima podatke preko komunikacijskih pinova koji dolaze sa RFID čitača. Koristi se kako bi napravio sistem pristupa koji kontrolira digitalni izlaz. Slika 9. prikazuje primjer RFID tag-ova i kartice.

MFRC522 je integrirani IC za beskontaktnu komunikaciju na 13.56 MHz. Ovaj čip upravlja antenom koja je dizajnirana za čitanje i slanje podataka. Bazna stanica, tj. RFID čitač sastoji se od antene, RF modula te kontrolne jedinice. RFID tagovi i kartice imaju sličnu konstrukciju, no bez aktivnih komponenti, što znači da ne sadrže nikakve baterije niti izvore napajanja.

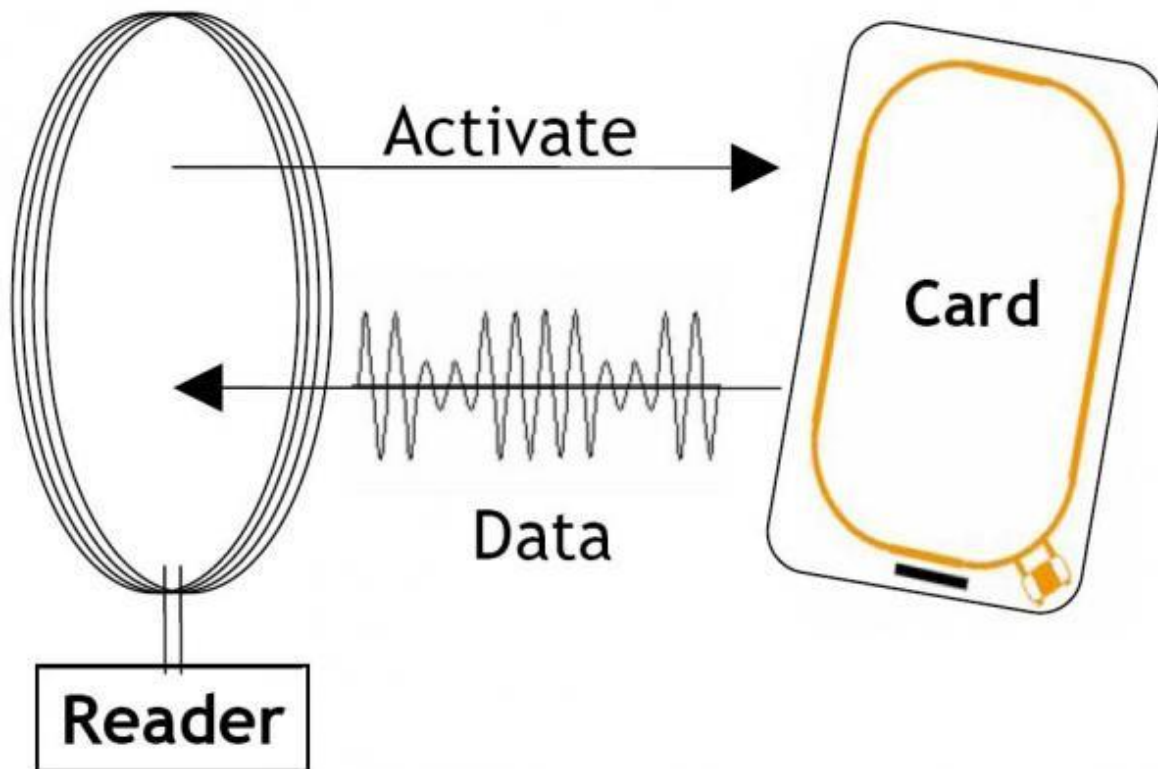
Sastoje se samo od antene i mikročipa. Komunikacija se odvija na sljedeći način. Bazna stanica (antena) konstantno odašilje elektromagnetske valove. Kada se RFID kartica ili tag približi bazi na udaljenost od 10-ak cm, utjecajem elektromagnetskog polja na prijemnoj anteni će se inducirati napon koji služi kao izvor napajanja ugrađenom mikročipu. Sada kada kartica ili tag imaju energiju potrebnu za normalan rad, ugrađeni čipovi manipuliraju teretom tako da bazna stanica može čitati niz jedinica i nula što je prikazano na slici [Slika 10.].



Slika 9. RFID čitač

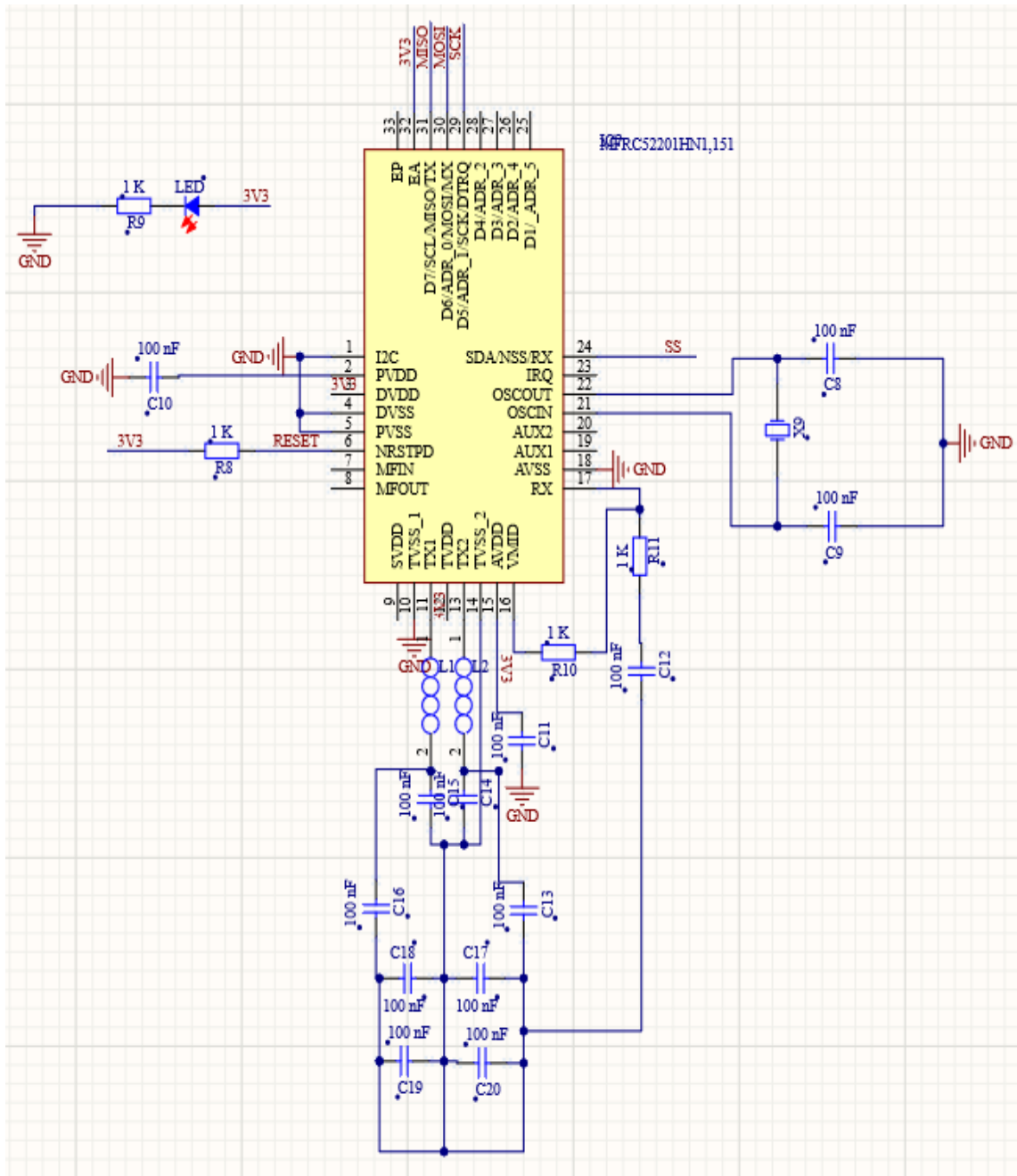
Tablica 3. Specifikacije RC522 čitača

Naziv čipa	MFRC522
Radna frekvencija	13,56 MHz
Ulazni napon	3,3 V
Jakost struje	30 mA
Udaljenost čitanja kartice	Oko 30 mm
Protokol	SPI
Dimenzije	40 mm x 60 mm
Maksimalna SPI brzina	10 Mb/s



Slika 10. Prikaz rada čitača

U svrhu izrade zasebne tiskane pločice na slici [Slika 11.] prikazan je shematski prikaz RC522 čitača.



Slika 11. Shematski prikaz RC522 čitača

3.4. Bluetooth HM-10

Modul Bluetooth HM-10 prikazan na slici [Slika 12.] omogućava Bluetooth 4.0(BLE 4.0) komunikaciju između projekta i drugog Bluetooth uređaja; npr. mobitela. Koristi se jer ima malu potrošnju, a lako se spaja na Arduino komunikacijske pinove. Dizajniran je za transparentno postavljanje bežične serijske veze. Komunikacija se odvija serijskom vezom što olakšava spajanje klijenta na računalo. Modul se može koristiti za slanje i primanje podataka.



Slika 12. Bluetooth modul HM-10

Bluetooth modul HM – 10 je vrlo svestran ima više verzija i može ga se koristiti u raznim kombinacijama kao što su : HM – 10 i mobilni uređaj, HM – 10 i HM – 10, HM – 10 i PC... itd. Ova verzija je montirana na breakout pločici koja izlaže napajanje i UART veze za pinove koji su prikladni za rad. Breakout pločica uključuje 3.3V regulator snage koji ih čini 5V kompatibilan. Te bi trebalo imati na umu da je RX pin još uvijek 3.3V i kada se koristi 5V Arduino treba pretvoriti Arduinov 5V TX na 3.3V za HM-10 RX.

Tablica 4. Specifikacije Bluetooth modula HM-10

Bluetooth standard	4.0
Funkcionira s	IOS6 i većim te Android 4.3 i većim
Napon	3V – 6V
Struja	60 uA - 400uA
Dimenzije	43 mm x 16 mm

Opis pinova:

STATE - Stanje konekcije

LOW - kada nije spojen. HIGH kada je spojen

VCC - Napajanje 3.6V do 6V

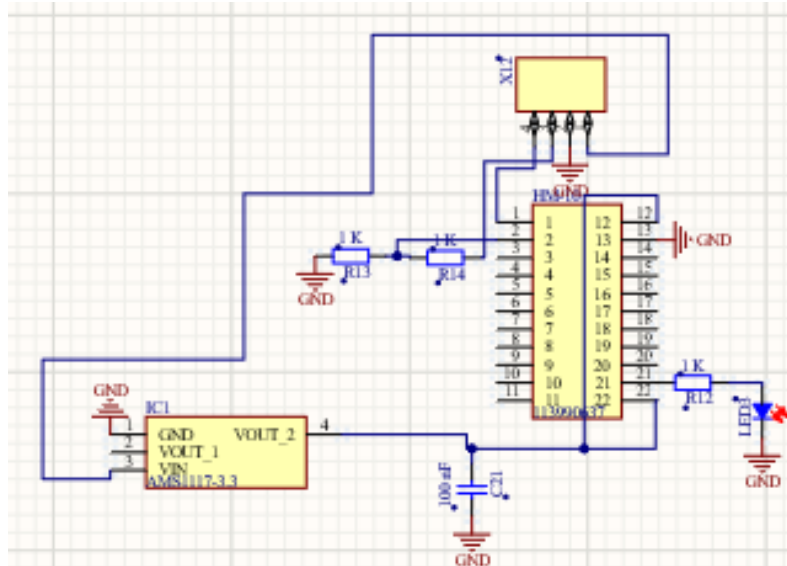
GND - Uzemljenje

TXD - Serial UART transmit

RXD - Serial UART receive

BRK - Break pin. Kada je konekcija aktivna, postavljajući BRK pin na LOW prekida konekciju. Kada nema konekcije postavljajući BRK pin na HIGH ili LOW nema nikakvog učinka.

Na slici [Slika 13.] prikazan je shematski prikaz Bluetooth modula koji će se kasnije koristiti za zasebnu tiskanu pločicu.



Slika 13. Shematski prikaz Bluetooth modula HM-10

3.5. *Senzor otiska prsta*

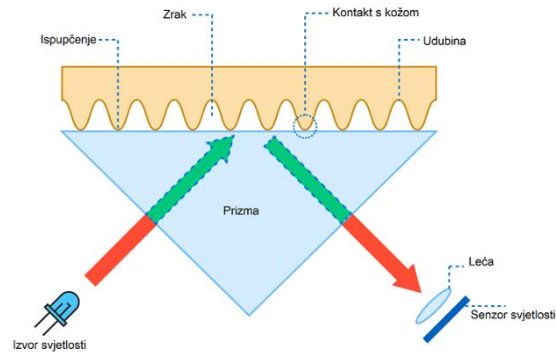
Senzor otiska prsta omogućuje čitanja i prepoznavanje otisaka prstiju. Sa mikroupravljačem komunicira preko serijske veze, a u mogućnosti je spremiti do 200 različitih otisaka prstiju. Nakon što se prisloni prst, fotografija otiska uspoređuje se s bazom spremljenih otisaka te ako isti prst postoji, daje se signal za otključavanje.

Ovaj senzor je optički što znači da pomoću svjetla prepoznaje otisak prsta, a osim optičkog ima još i kapacitivni senzor(koristi se na mobilnim uređajima) i ultrazvučni. Optički senzor otiska prsta je starija tehnologija pa je stoga i jednostavnija od ostalih, ali baš zbog jednostavnosti rada je pristupačan.



Slika 14. Senzor otiska prsta

Kao što znamo na prstu imamo ispupčenja i udubine koje su jedinstvene za svaku osobu i to je otisak prsta. Da bi detektirao udubine i ispupčenja senzor otiska prsta koristi led diode kao osvjetljenje i fotodetektor kako bi primio odbijene zrake svjetlosti. Naravno senzor nema samo jednu diodu i fotodetektor nego se unutar njega nalazi puno fotodetektora smještenih u matricu, kao i prizma i leća kako bi točnije mogao primiti odbijenu svjetlost. Na slici [Slika 15.] vidimo princip na koji senzor radi i koji je prethodno objašnjen.



Slika 15. Princip rada senzora otiska prsta

Naravno u senzoru imamo i kontroler koji upravlja diodama i fotodetektorima te analizira primljene signale. Kako bi mogao prepoznati otisak kontroler u memoriji ima bazu spremljenih otisaka te ih uspoređuje s trenutnim otiskom i ako ga ima u bazi vraća nam koliko se podudaraju.

Tablica 5. Specifikacije senzora otiska prsta

Napon	3,3 V
Struja	120 mA
Komunikacija	UART
Dimenzije	50 x 21 x 20 mm

Senzor ima 6 pinski kabel, ali za spajanje se koriste samo 4 pina (2 za napajanje i 2 za komunikaciju). RX pin se spaja na D11, a TX sa senzora spaja se na D10 na Arduinu, za napajanje se koristi 3,3V kako je naznačeno na senzoru.

3.6. Elektromagnetska brava za vrata

Elektromagnetska brava prikazana na slici [Slika 16.] omogućava zaključavanje vrata putem mikroupravljača. Specifikacije brave prikazane su u tablici [Tablica 6.]. Navedena brava može držati do 80kg svojim elektromagnetom, što je otprilike na granici za korištenje s pravim vratima. Aktiviran je, tj. zaključava se kada mu se dovede struja. Sastoji se od dva dijela: elektromagneta i komada metala. Ugrađuje se vijcima za ugradnju.



Slika 16. Elektromagnetska brava

Tablica 6. Specifikacije elektromagnetske brave

Napon aktivacije	12 V
Stalna struja aktivacije	100 mA
Dimenzije	80 x 40 x 25 mm
Težina	Cca 500 g

3.7. Relej

Relej je elektromagnetski prekidač pomoću kojeg možemo s manjim naponom kontrolirati protok struje većeg napona. Modul prikazan na slici [Slika 17.] koristi se za aktiviranje elektromagnetske brave na principu - uključivanjem releja pokrenuti će se brava što će omogućiti ulaz korisniku.



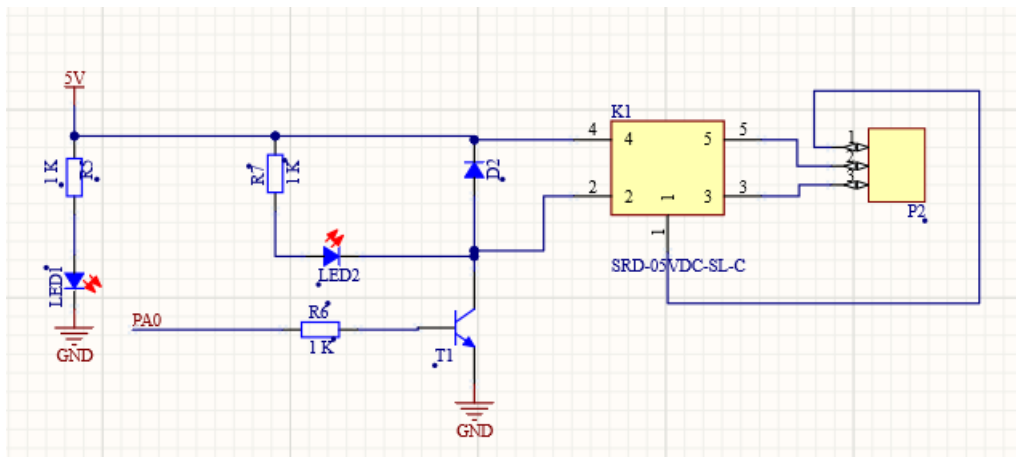
Slika 17. Relej

Elektromagnetski relej se sastoji od zavojnice, željezne kotve i kontakta. Kada struja proteče kroz zavojnicu, stvara se magnetsko polje koje privlači željeznu kotvu. Na kotvi se nalaze kontakti koji ovisno o protoku struje zatvaraju ili otvaraju drugi strujni krug. Unutar tog drugog strujnog kruga moguće je spojiti trošilo koje zahtjeva visoki napon ili izmjenični napon. Za rukovanje relejom potrebno je znati nekoliko karakteristika samog releja. Prvo je količina i vrsta napona koju je potrebno dovesti zavojnici da se aktivira. Postoje razne vrijednosti tog napona ovisno o potrebi i namjeni (često 5, 12, 24V DC). Druga stvar na koju je potrebno obratiti pažnju je snaga trošila koje spajamo na relej. Na samom releju je napisana maksimalna količina struje i napona koja može biti spojena s relejom (često 12, 14, 30V DC ili 120, 220V AC). Ako se prekorače granice, relej se može oštetiti ili pregorjeti.

Tablica 7. Specifikacije releja

Struja potrebna za aktivaciju	20 mA
Napon zavojnice	5 V
Max DC	12 V, 10 A
Max AC	250 10 A

U svrhu izrade zasebne tiskane pločice na slici [Slika 18.] prikazan je shematski prikaz releja.



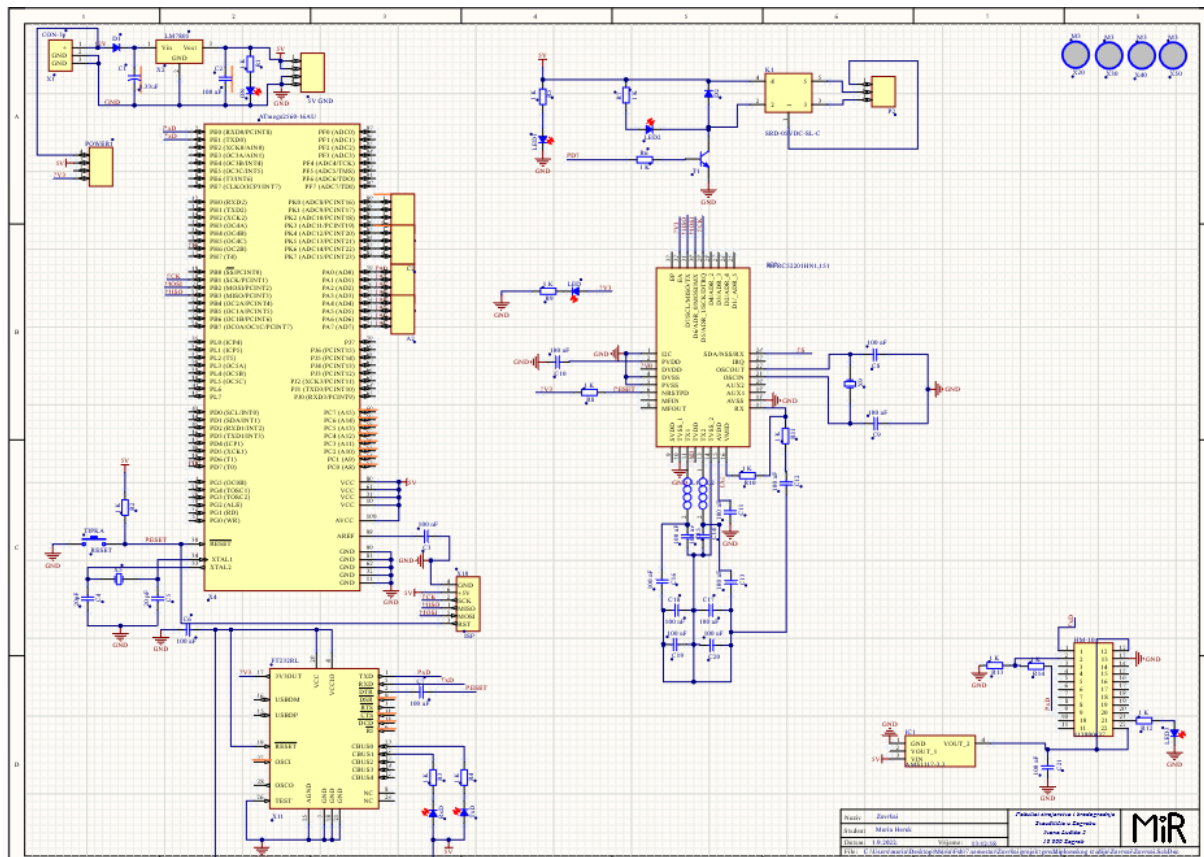
Slika 18. Shematski prikaz releja

4. SPAJANJE I TESTIRANJE

Nakon odabira gotovih modula koji će se koristiti za izradu sklopa elektroničke brave potrebno je međusobno ih spojiti na odgovarajuće priključke svakog od uređaja. Prvi korak je pažljivo proučiti svaki priručnik (engl. datasheet) od pojedinog modula, vidjeti protokol kojim komunicira, te zatim taj protokol i priključke pronaći na Arduino Mega 2560 pločici i spojiti ih.

4.2. Električna shema

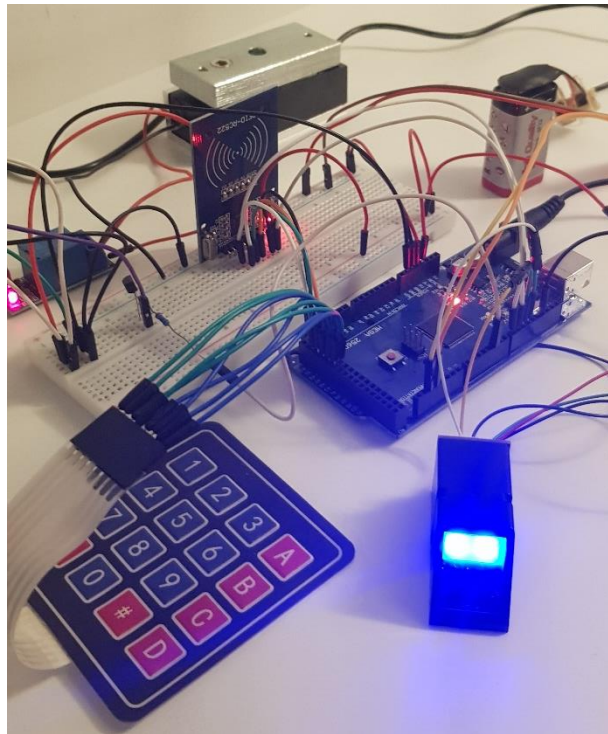
Električna shema, koja je korištena u svrhu izrade zasebne tiskane pločice, nacrtana je i prikazana je na slici [Slika 19.].



Slika 19. Električna shema

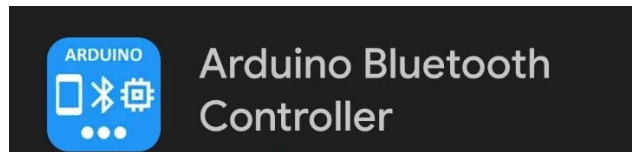
4.3. Eksperiment

Svaku teoriju potrebno je potvrditi eksperimentom. Potrebno je provjeriti ispravnost komponenata, spojeva i njihova funkcionalnost. Također prije samog spajanja potrebno je programirati mikrokontroler kako bismo odredili protokol komunikacije prema ostalim modulima i definirali sve potrebne funkcije. Taj dio obavljen je u sljedećem poglavlju. Nakon programiranja u svrhu potvrde teorije eksperimentom koristila se eksperimentalna ploča na kojoj su pomoću žica za spajanje i već ranije spomenutih modula prema ranije navedenoj shemi [Slika 19] spojeni svi elementi u cjelinu. Prikazani eksperimentalni spoj može se vidjeti na slici [Slika 20]. Ovime je potvrđen rad samog sklopa nakon dovođenja napajanja na samu Arduino pločicu koji će detaljnije biti opisan u poglavlju programiranja (poglavlje 5).



Slika 20. Eksperimentalni spoj

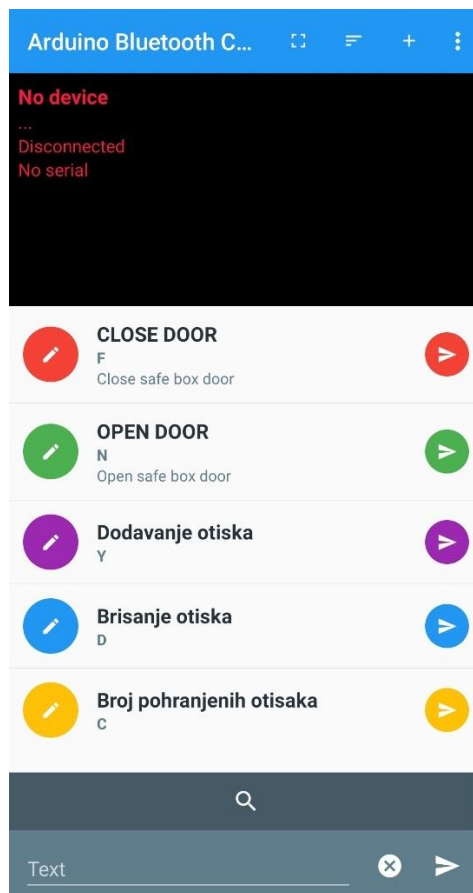
Za upravljanje radom preko pametnog telefona koristi se aplikacija pod nazivom Arduino Bluetooth Controller koja je dostupna na Trgovini Play za skinuti. Ikona aplikacije prikazana je na slici [Slika 21].



Slika 21. Ikona aplikacije za pametni telefon

Sučelje aplikacije prikazano je na slici [Slika 22]. Sučelje se po vlastitoj želji može mijenjati.

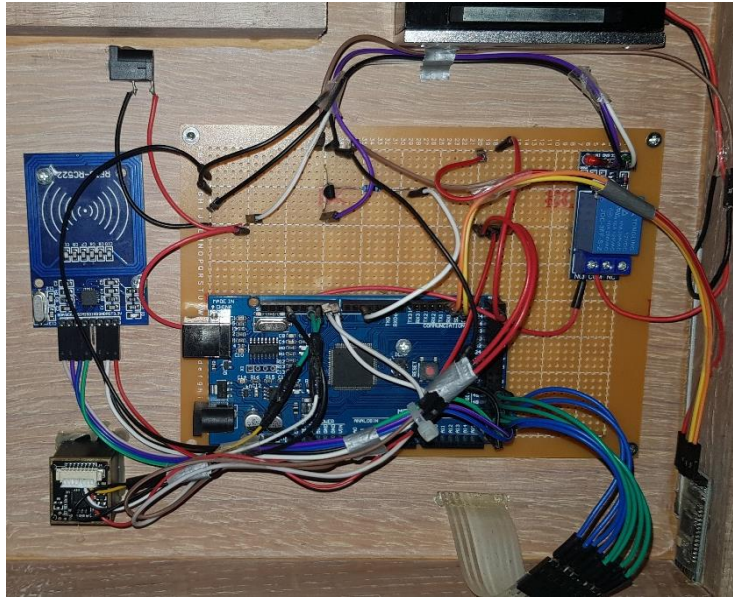
Ona služi za otvaranje i zatvaranje vrata te kao što je prikazano na slici za dodavanje, brisanje i prikazivanje broja pohranjenih otisaka.



Slika 22. Sučelje aplikacije

4.4. Izrada prototipne pločice

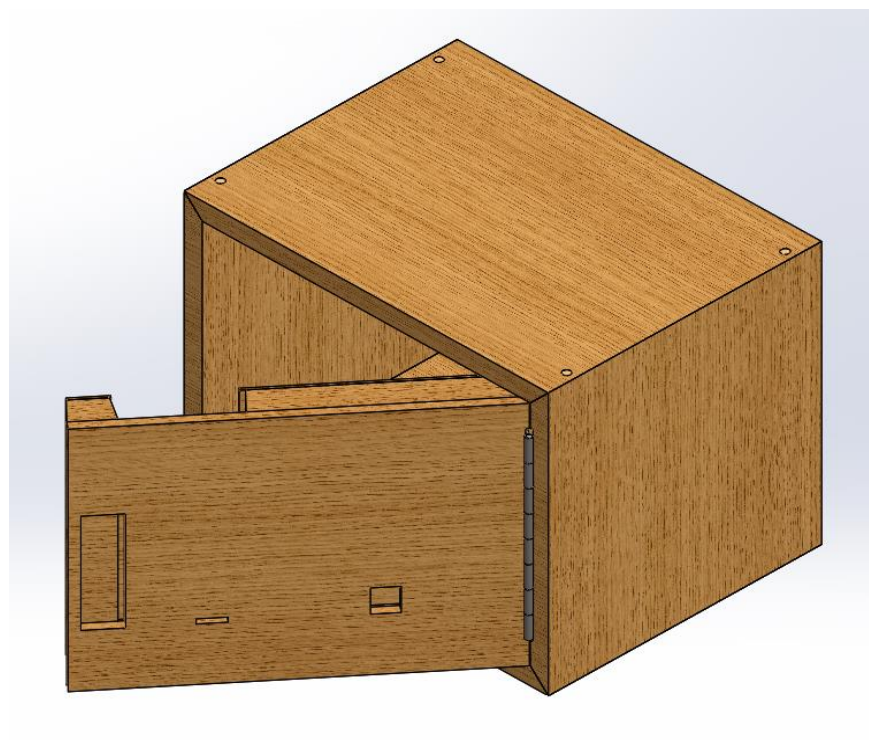
U izradi PCB pločice korištena je pločica s rupicama. Takav pristup ne zahtijeva izradu tiskane pločice čija cijena izrade u malim serijama, u ovom slučaju jednog komada, može biti visoka.



Slika 23. Prototipna pločica

Nakon izrade prototipne pločice potrebno je za nju izraditi kućište koje je prvo konstruirano u programu SolidWorks (verzija 25) čiji je model prikazan na slici [Slika 24.]. SolidWorks je CAD program za crtanje 3D modela na Microsoft Windows računalu. Omogućuje modeliranje dijelova i spremanje na mnogo različitih načina.

Gotovo kućište prikazano je slikom [Slika 25.]. Izvedeno je drveno kućište koje je cjenovno povoljnije i prikazuje zapravo prototip sefa.



Slika 24. Model kućišta



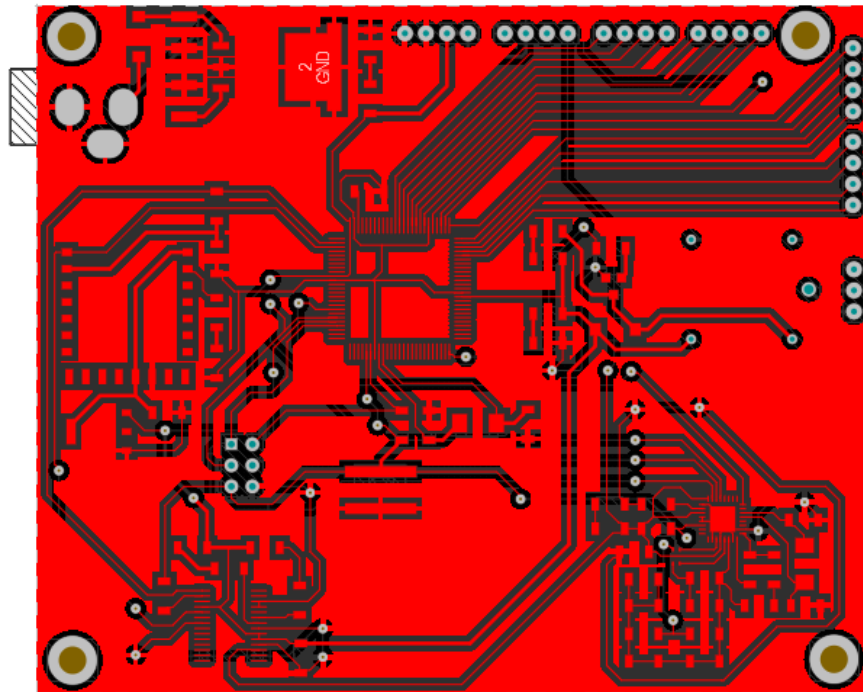
Slika 25. Kućište

4.5. Alternativna izrada PCB pločice

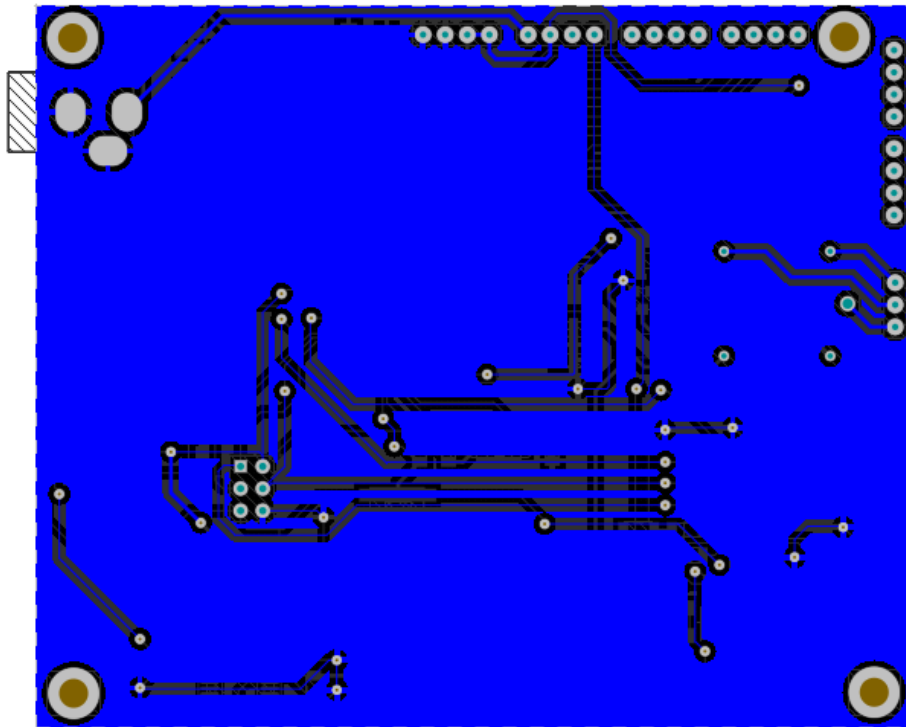
Za takav način izrade potrebno je bilo nacrtati PCB (engl. *Printed Circuit Board*) datoteku u jednom od programa za to. Za crtanje shematskih prikaza pojedinih modula, električne sheme, a naknadno i PCB-a, korišten je programski alat koji se naziva Altium Designer (verzija 20.0.13). Ovaj program sadrži širok izbor stvarnih komponenata proizvedenih od strane etabliranih proizvođača. Također postoji i online knjižnica u kojoj je moguće pronaći za komponente odgovarajuće 3D modele.

On je izumljen 2005. godine od Australске kompanije koja se naziva Altium Limited. Taj program se jako brzo proširio po cijelom svijetu i danas je jedan od najboljih alata za projektiranje tiskanih pločica.

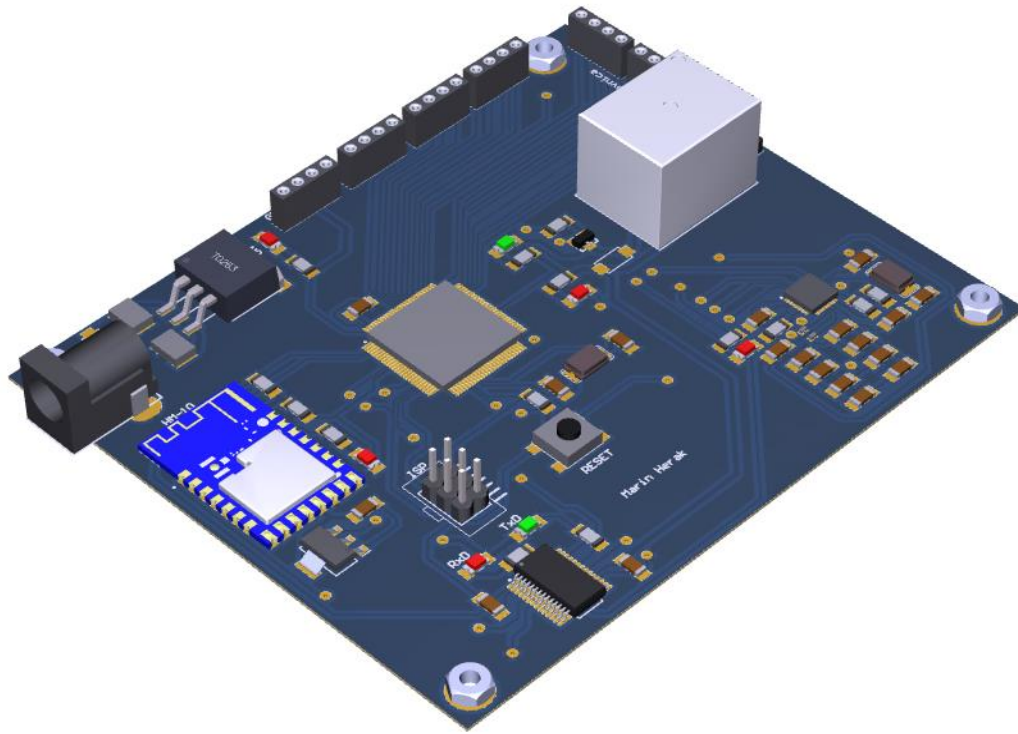
PCB je spajan ručno i prikazan je sljedećim slikama na različite načine.



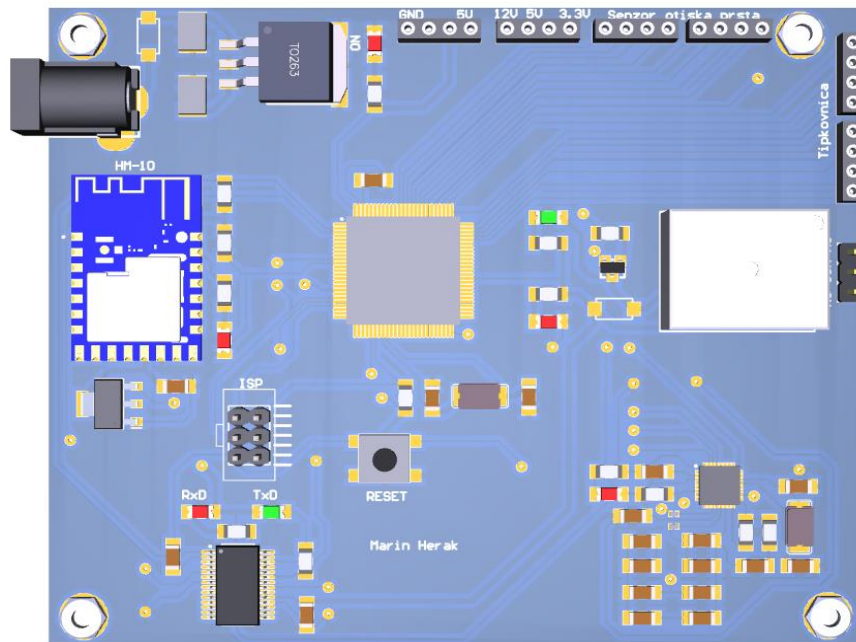
Slika 26. Top layer PCB-a



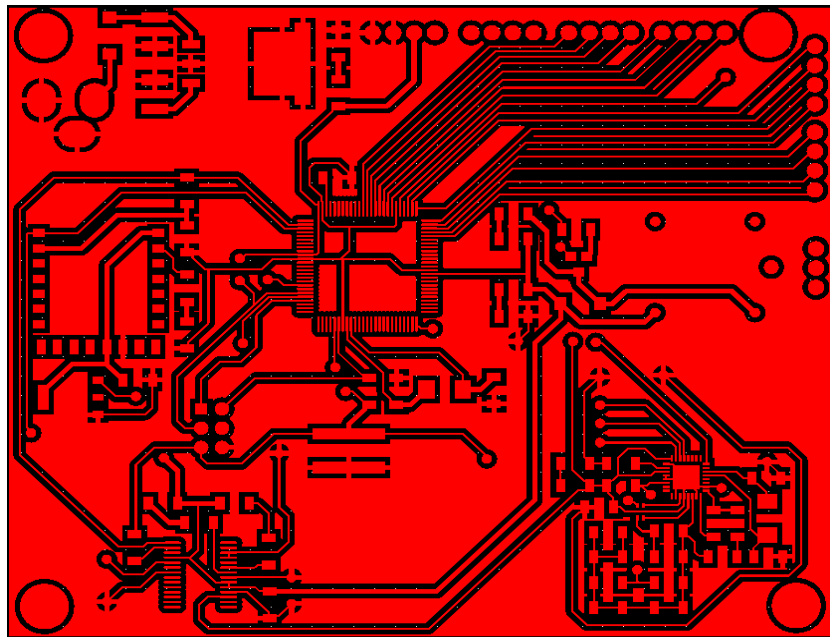
Slika 27. Bottom layer PCB-a



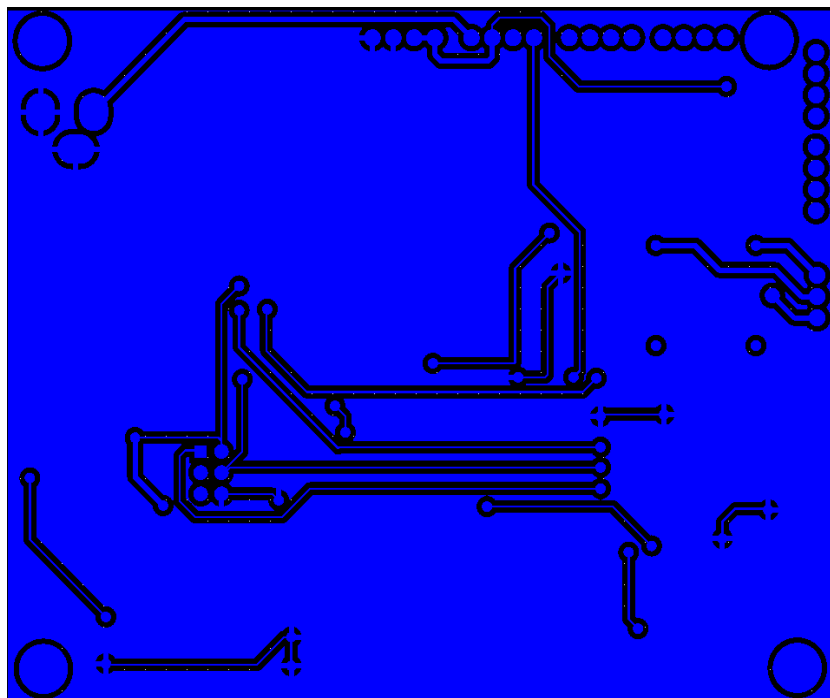
Slika 28. 3D prikaz



Slika 29. 3D prikaz odozgo



Slika 30. Gerber file top layer-a

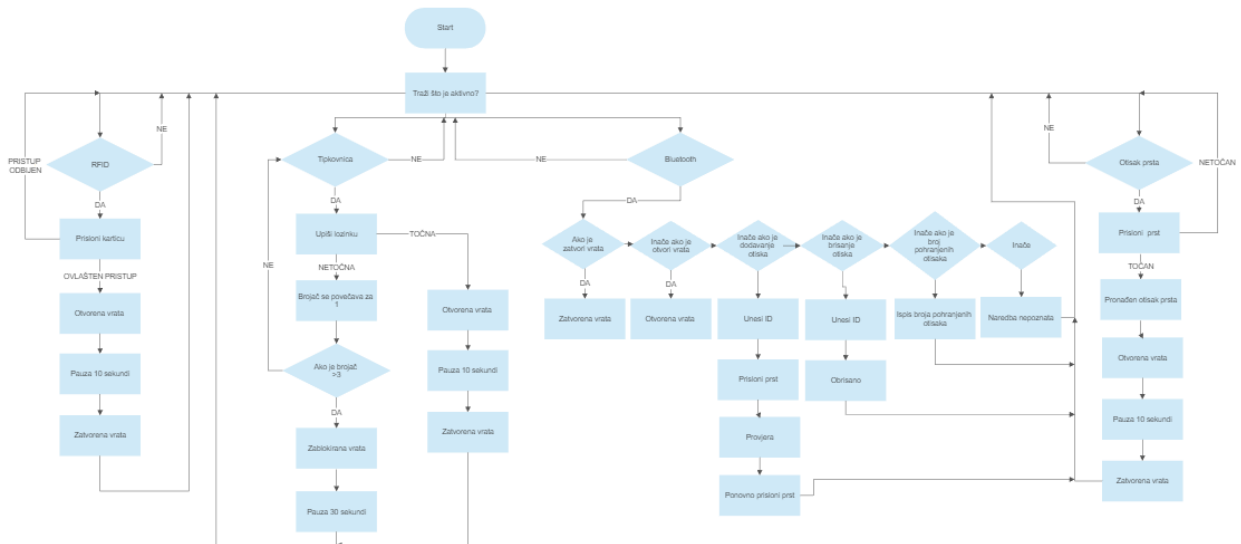


Slika 31. Gerber file bottom layer-a

5. PROGRAMIRANJE

5.1. Dijagram toka

Osmišljen je dijagram toka odnosno logika kojom će program izvoditi naredbe prikazana na slici[Slika 32].



Slika 32. Dijagram toka

5.2. Arduino IDE

Programski kod napisan je u Arduino IDE (engl. *Integrated Development Enviroment*) sučelju. Sučelje se sastoji od prozora za uređivanje teksta, trake s porukama, konzole i alatne trake s nizom izbornika. U njemu se može napisati kod za Arduino pločice, ali uz određene prilagodbe i za neke druge. Programski kod se temelji na C++ programskom jeziku. Aktualna verzija programa je 1.8.19.



```
Automatska_sigurnosna_vrata | Arduino 1.8.13
File Edit Sketch Tools Help
Automatska_sigurnosna_vrata
1 //*****Marin Herak*****//
2 /**Automatska sigurnosna vrata**//
3
4 #include <Keypad.h>           //Include the Keypad library
5 //#include <SoftwareSerial.h> //Include the SoftwareSerial library
6 #include <SPI.h>             // Include the standard Arduino SPI lib:
7 #include <MFRC522.h>         // Include the RFID library
8 #include <Adafruit_Fingerprint.h> //Include the library for fingerpr:
9
10 //KEYPAD and RELAY
11 const int RELAY_PIN = 5; // the Arduino pin, which connects to the I
12 const int ROW_NUM = 4; //four rows
13 const int COLUMN_NUM = 4; //four columns
14
15 char keys[ROW_NUM][COLUMN_NUM] = {
16   {'1', '2', '3', 'A'},
17   {'4', '5', '6', 'B'},
18   {'7', '8', '9', 'C'},
19   {'*', '0', '#', 'D'}
20 };
21
22 byte pin_rows[ROW_NUM] = {48, 46, 44, 42}; //connect to the row pinout
23 byte pin_column[COLUMN_NUM] = {49, 47, 45, 43}; //connect to the colu
24 char key;
```

Slika 33. Arduino IDE sučelje

Kod napisan u Arduino IDE sučelju sastoji se od 3 dijela.

Prvi dio koda, prije inicijalizacijske rutine (engl. *setup*), sastoji se najčešće od uključivanja dodanih library-a u kod i deklariranja korištenih varijabli te definiranja nekih osnovnih parametara, ovisno o primjeni.

Drugi dio koda upisuje se unutar void *setup()* funkcije. Tako napisan kod izvršava se samo jednom prilikom pokretanja Arduina.

Treći dio koda upisuje se unutar void loop() funkcije i izvršava se isponova u petlji cijelo vrijeme dok Arduino radi sve do njegovog gašenja.

5.3. Deklaracija varijabli

```
#include <Keypad.h>           //pozivanje Keypad biblioteke
#include <SPI.h>               // pozivanje standardne Arduino SPI biblioteke
#include <MFRC522.h>          // pozivanje RFID biblioteke
#include <Adafruit_Fingerprint.h> //pozivanje biblioteke za senzor otiska prsta

//KEYPAD and RELAY
const int RELAY_PIN = 5; // Arduino pin, IN pin releja
const int ROW_NUM = 4; //4 reda
const int COLUMN_NUM = 4; //4 stupca

char keys[ROW_NUM][COLUMN_NUM] = { //definiranje tipki tipkovnice
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte pin_rows[ROW_NUM] = {48, 46, 44, 42}; //spajanje pinova redova tipkovnice
byte pin_column[COLUMN_NUM] = {49, 47, 45, 43}; //spajanje pinova stupaca tipkovnice
char key;
int brojac;

Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM );
```

```
const String password_1 = "1234ABC"; // lozinka
String input_password;

//RFID
#define SDA_DIO 9
#define RESET_DIO 8 // definiranje SDA(SS) i RST (reset ) pinove
MFRC522 RC522(SDA_DIO, RESET_DIO); // kreiranje instance RFID biblioteke
boolean RFIDMode = true; //boolean za mijenjanje modova

//BLUETOOTH
#define softserial
#ifndef softserial
#include <SoftwareSerial.h> //pozivanje SoftwareSerial biblioteke
  SoftwareSerial HM10(19, 18); // RX = 10 ili 19, TX = 11 ili 18,
#else
  HardwareSerial HM10 = Serial1;
#endif

String inData ;

//FINGERPRINT
SoftwareSerial mySerial(10, 11); //(RX, TX) (10(RX) sa Arduino Mega2560 spajamo na TX
senzora(zuta), a 11(TX) sa Arduino Mega2560 na RX senzora(bijela)

//Konstruktor za Adafruit_Fingerprint biblioteku kojemu predajemo Serijski port na koji je spojen senzor(
u našem slučaju je to Softverski serijski port)

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

//varijabla u koju spremamo ID pod kojim korisnik želi spremiti novi otisak prsta
uint8_t id;
```

5.4. Setup

U inicijalizacijskoj (setup) rutini potrebno je definirati što će se sve izvršiti prilikom pokretanja. Najprije treba napisati kod koji će omogućiti prijenos podataka putem serijske komunikacije. Potom treba definirati koji pinovi su ulazni, a koji izlazni te njihove inicijalne vrijednosti.

```
Serial.begin(9600); //označavanje početka serijske komunikacije
SPI.begin();
Serial1.begin(9600); // setiranje HM10 na 9600
finger.begin(57600); //Inicijalizacija komunikacije sa senzorom
Serial1.println("Bluetooth veza je uključena");
RC522.PCD_Init(); // inicijalizacija RFID-a
input_password.reserve(32); // maksimalni broj znakova je 32
pinMode(RELAY_PIN, OUTPUT); // inicijalizacija pina kao izlaznog
digitalWrite(RELAY_PIN, HIGH); // zaključaj vrata
pinMode(13, OUTPUT); // inicijalizacija pina kao izlaznog
```

5.5. Petlja

U prvom dijelu petlje traži se koji od dijelova je aktivan, odnosno na koji način želimo otvoriti vrata, i zatim se poziva određena funkcija.

```
//Look for press key
key = keypad.getKey();
if (key) {
  Serial1.print("* ");
  getKeypad();
}
```

```
//Look for Bluetooth
if (Serial1.available()){
  getBluetooth();
}
//Look for RFID
if(RFIDMode == true){
  getRfid();
}
// Look for fingerprint
getFingerprintIDez(); //funkcija u kojoj očitavamo otisak prsta, ako je prst prislonjen na senzor
delay(50);
}
```

Ako je aktivna tipkovnica poziva se sljedeća funkcija u kojoj se provjerava je li ispravna lozinka ili nije. Ako je lozinka ispravna vrata se otvaraju na 10 sekundi, a ako nije brojač se povećava za 1. Ako je lozinka neispravna 3 puta onda se vrata prisilno zaključavaju na 30 sekundi.

```
void getKeypad(){
  if (key == '*') {
    input_password = ""; // resetiranje ulazne lozinke
  } else if (key == '#') {
    if (input_password == password_1) {
      Serial1.println("Lozinka je ispravna");
      OpenDoor10();
    }
  } else {
    Serial1.println("Lozinka je neispravna, pokušaj ponovno");
    brojac=brojac+1;
  }
}
```



```
input_password = ""; // resetiranje ulazne lozinke
} else {
input_password += key; // dodavanje novog znaka ulaznoj lozinki
    }
if (brojac==3){
    CloseDoorBreakIn(); // funkcija pokušaj provale
    }
}
```

Ako je aktivan RFID provjerava se je li pristup ovlašten ili odbijen ovisno o prislonjenoj kartici.

```
void getRfid(){
    // Look for new cards
    if ( ! RC522.PICC_IsNewCardPresent()
    {
        return;
    }
    // Select one of the cards
    if ( ! RC522.PICC_ReadCardSerial()
    {
        return;
    }
    //Show UID on serial monitor
    String content= "";
    byte letter;
```

```
for (byte i = 0; i < RC522.uid.size; i++)
{
    Serial.print(RC522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(RC522.uid.uidByte[i], HEX);
    content.concat(String(RC522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(RC522.uid.uidByte[i], HEX));
}

content.toUpperCase();

if (content.substring(1) == "0C 78 CB 6E") //change here the UID of the card/cards that you want to give
access
{
    Serial1.println("Ovlašten pristup");
    Serial1.println("Dobrodošli Marin Herak");
    OpenDoor10();
}

else if (content.substring(1) == "B3 E7 16 04") //change here the UID of the card/cards that you want to
give access
{
    Serial1.println("Ovlašten pristup");
    Serial1.println("Dobrodošli Pero Perić");
    OpenDoor10();
}
```

```
else {  
    Serial.println("Pristup odbijen");  
    CloseDoor();  
}  
}  
  
//funkcija koja čita broj sa i vraća taj broj  
uint8_t readnumber(void) {  
    uint8_t num = 0;  
  
    while (num == 0) {  
        while (! Serial1.readString());  
        num = Serial1.parseInt();  
    }  
    return num;  
}
```

Kod aktivnog Bluetooth-a može se otvarati i zatvarati vrata te dodavati, brisati i prikazivati broj pohranjenih otisaka. Potrebno je samo pratiti poruke na zaslonu i odabrati željenu naredbu.

```
void getBluetooth(){  
    inData = Serial1.readString();  
    Serial1.println(inData);  
    if ( inData == "F") {  
        CloseDoor();  
    }  
}
```

```
else if ( inData == "N") {
    OpenDoor();
}
else if(inData == "Y"){
    verifyFingerprint();
    Serial1.println("\nSpreman za dodavanje otiska!");
    Serial1.println("Unesite ID#(od 1 do 127) pod kojim zelite spremiti novi otisak prsta...");
    id = readnumber();
    if (id == 0) { //ako je uneseni broj 0 vraćamo se na početak i tražimo ponovno unošenje broja
        return;
    }
    Serial1.print("Upisivanje ID #");
    Serial1.println(id);
    //pozivamo funkciju koja obavlja unošenje novog otiska
    while (! getFingerprintEnroll() );
}
else if(inData == "D"){
    verifyFingerprint();
    Serial1.println("Unesite ID # otiska (od 1 do 127) koji želite obrisati...");
    uint8_t id = readnumber();//u varijablu id spremamo broj koji nam vraća funkcija readnumber koja čita broj sa Serial monitora
    if (id == 0) { // ako je id nula vraćamo se na početak te ponovno čekamo unos broja
        return;
    }
    Serial1.print("Brisanje ID #");
    Serial1.println(id);//ispisujemo koji ID se briše
    //pozivamo funkciju koja obavlja unošenje novog otiska
    deleteFingerprint(id);// pozivamo funkciju za brisanje otiska
}
```

```
else if(inData == "C"){
    finger.getTemplateCount();//funkcija za dohvaćanje podataka o broju spremljenih otisaka
    //ispisujemo broje spremljenih otisaka na Serial monitor
    Serial1.print("Senzor ima "); Serial1.print(finger.templateCount); Serial1.println(" spremljenih
otisaka");
}
else {
    Serial1.println("Naredba nije poznata!");
}
}
void verifyFingerprint(){
    //provjeravamo lozinku senzora koja je postavljena u biblioteci( može se promijeniti u kodu)
    //ako je lozinka točna( ako nismo mijenjali onda će biti točna) ispisujemo poruku da je senzor pronađen(
ako smo provjerili lozinku znači da je i komunikacija sa senzorom uspjela)
    if (finger.verifyPassword()) {
        Serial1.println("Pronadjen senzor otiska prsta!");
    }
    else { //ako je lozinka netočna ili nije pronađen senzor ispisujemo poruku na serial monitor ostajemo u
beskonačnoj while petlji
        Serial1.println("Senzor otiska prsta nije pronađen");
        while (1) while (1) { delay(1); }
    }
}

uint8_t getFingerprintEnroll() {

    int p = -1;
    Serial1.print("Čekam novi otisak prsta s ID #"); Serial1.println(id);
    delay(1000);
    //sve dok ne dobijemo ispravan otisak ponavljamo postupak slikanja
```

```
while (p != FINGERPRINT_OK) {  
  p = finger.getImage();// uzimamo sliku sa senzora otiska i u varijablu p spremamo povratnu vrijednost  
  //prema vrijednosti u varijabli p ispisujemo odgovarajuću poruku što je senzor napravio, odnosno dali se  
  dogodila greška  
  switch (p) {  
    case FINGERPRINT_OK:  
      Serial1.println("Slika napravljena");  
      break;  
    case FINGERPRINT_NOFINGER:  
      Serial1.print(".");  
      delay(100);//pauza da se . ne ispisuje stalno  
      break;  
    case FINGERPRINT_PACKETRECEIVEERR:  
      Serial1.println("Komunikacijska greska");  
      break;  
    case FINGERPRINT_IMAGEFAIL:  
      Serial1.println("Greska sa slikom");  
      break;  
    default:  
      Serial1.println("Nepoznata greska");  
      break;  
  }  
}
```

```
p = finger.image2Tz(1);//snimljenu sliku pretvaramo u drugi format koji se može pohraniti u memoriju  
switch (p) {  
  //prema vrijednosti u varijabli p ispisujemo dali je pretvaranje uspjele ili se dogodila greška  
  case FINGERPRINT_OK:  
    Serial1.println("Slika pretvorena");
```

```
break;
case FINGERPRINT_IMAGEMESS:
Serial1.println("Slika neispravna");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial1.println("Komunikacijska greška");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial1.println("Nemoguće pronaći značajke otiska");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial1.println("Nemoguće pronaći značajke otiska");
return p;
default:
Serial1.println("Nepoznata greška");
return p;
}

// korisnik treba maknuti prst sa senzora te onda ponovno staviti kako bi potvrdili otisak
Serial1.println("Maknite prst sa senzora");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
Serial1.print("ID "); Serial.println(id);
p = -1;
Serial1.println("Stavite isti prst ponovno na senzor");

//kada korisnik ponovno stavi prst ponavljamo slikanje i konverziju slike te ispisujemo poruke ako se
dogodi greška kao i u prethodnom slučaju
```

```
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial1.println("Slika napravljena");
break;
case FINGERPRINT_NOFINGER:
Serial1.print(".");
delay(100);//pauza da se . ne ispisuje stalno
break;
case FINGERPRINT_PACKETRECEIVEERR:
Serial1.println("Komunikacijska greška");
break;
case FINGERPRINT_IMAGEFAIL:
Serial1.println("Greška sa slikom");
break;
default:
Serial1.println("Nepoznata greška");
break;
}
}
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
Serial1.println("Slika pretvorena");
break;
case FINGERPRINT_IMAGEMESS:
Serial1.println("Slika neispravna");
```



```
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial1.println("Komunikacijska greška");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial1.println("Nemoguće pronaći značajke otiska");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial1.println("Nemoguće pronaći značajke otiska");
return p;
default:
Serial1.println("Nepoznata greška");
return p;
}

//Ispisujemo korisniku da je obrada slike gotova i da se izrađuje model otiska
Serial1.print("Pravim model za #"); Serial1.println(id);

p = finger.createModel();//funkcija koju pozivamo kako bi napravili model otiska prsta
// provjeravamo povratnu vrijednost funkcije kako bi znali što je napravljeno
if (p == FINGERPRINT_OK) {
Serial1.println("Otisci se poklapaju!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial1.println("Komunikacijska greska");
return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
Serial1.println("Otisci se ne poklapaju");
return p;
```

```
    } else {
        Serial1.println("Nepoznata greška");
        return p;
    }
    //ako se otisci poklapaju onda spremamo model otiska u memoriju senzora
    Serial1.print("ID "); Serial1.println(id);
    p = finger.storeModel(id);//funkcija za spremanje modela otiska u memoriju senzora koja nam vraća jeli
    spremanje uspelo ili ne
    //Ispisujemo poruku ovisno o povratnoj vrijednosti funkcije za spremanje
    if (p == FINGERPRINT_OK) {
        Serial1.println("Spremljeno!\n");
    } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
        Serial1.println("Komunikacijska greška");
        return p;
    } else if (p == FINGERPRINT_BADLOCATION) {
        Serial1.println("Nemoguće spremiti na toj lokaciji ( probajte drugi ID)");
        return p;
    } else if (p == FINGERPRINT_FLASHERR) {
        Serial1.println("Greška sa spremanjem u memoriju");
        return p;
    } else {
        Serial1.println("Nepoznata greška");
        return p;
    }
}
```

Ako se prisloni otisak prsta izvodi se sljedeće:

```
//funkcija u kojoj očitavamo otisak i koja nam vraća -1 ako se dogodila greška ili ID otiska prsta kao se
podudara s nekim od spremljenih otisaka
```

```
void 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;

Serial1.print("Pronađen ID #");
Serial1.print(finger.fingerID); //ispisujemo ID otiska prsta
Serial1.print(" s pouzdanjem ");
Serial1.println(finger.confidence); //ispisujemo koliko je pouzdan pročitan otisak s određenim otiskom iz
baze

OpenDoor10();
return finger.fingerID; //funkcija vraća ID otiska prsta
}

// funkcija za brisanje otisaka
uint8_t deleteFingerprint(uint8_t id) {
    uint8_t p = -1;

    p = finger.deleteModel(id); //brišemo otisak s predanim ID brojem, a funkcija nam vraća rezultat brisanja

    //ovisno što nam je prethodna funkcija vratila( uspješno brisanje ili pogrešku) ispisujem poruku ovisno
    što se dogodilo
    if (p == FINGERPRINT_OK) {
```

```
Serial1.println("Obrisano!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial1.println("Komunikacijska greška");
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial1.println("Nije moguće obrisati na toj lokaciji");
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial1.println("Greška pisanja po memoriji");
return p;
} else {
Serial1.print("Nepoznata greška: 0x"); Serial1.println(p, HEX);
return p;
}
}

//funkcija za otključavanje vrata na 10 s
void OpenDoor10(){
    digitalWrite(RELAY_PIN, LOW); // otključavanje vrata
    Serial1.println("Otvorena vrata na 10 sekundi");
    delay(10000); //pauza 10 s
    digitalWrite(RELAY_PIN, HIGH); //zaključavanje vrata
    Serial1.println("Zatvorena vrata");
}

void CloseDoorBreakIn(){
    Serial1.println("Pokušaj provale");
    digitalWrite(RELAY_PIN, HIGH); // zaključavanje vrata
    Serial1.println("Zatvorena vrata na 30 sekundi");
```

```
Serial1.print("30...");delay(15000);Serial1.print("15...");delay(10000);Serial1.print("5...");delay(1000);  
Serial1.print("4...");delay(1000);Serial1.print("3...");delay(1000);Serial1.print("2...");delay(1000);  
Serial1.println("1...");delay(1000);  
brojac=0;  
}
```

```
//funkcija za otključavanje
```

```
void OpenDoor(){  
    digitalWrite(RELAY_PIN, LOW); // otključavanje vrata  
    Serial1.println("Otvorena vrata");  
    delay(1000);  
}
```

```
//funkcija za zaključavanje
```

```
void CloseDoor(){  
    digitalWrite(RELAY_PIN, HIGH); // zaključavanje vrata  
    Serial1.println("Zatvorena vrata");  
    delay(1000);  
}
```

6. ZAKLJUČAK

Posjedovanje elektroničke brave olakšava ljudsku svakodnevicu. Bez narušavanja sigurnosti prilikom ulaska kroz vrata zadržava se karakteristika originalnog ključa za otključavanje vrata uz povećanu zaštitu zbog same enkripcije podataka bluetooth uređaja.

Cilj završnog rada bio je složiti automatska vrata sa sigurnosnim otključavanjem na više različitih načina. Pomoću Arduino Mega pločice uspješno je realiziran sklop te se omogućio pristup na više načina. Za Arduino Mega pločicu kreiran je programski kod koji će u odnosu na izlazni signal odraditi otključavanje, to jest zaključavanje. Uloga Bluetooth-a je najvećim djelom upravljanje sklopom i davanje povratnih informacija korisniku.

Najveći nedostatak opisanog sklopa jest upravo softverski i elektronički dio koji je danas moguće hakirati (obično se kaže da ne postoji zaštita koja se ne može probiti) i neovlašteno zloupotrebljavati.

Prednosti su već nekoliko puta spomenute kroz ovaj rad, ali glavna je jednostavno i brže otvaranje i zatvaranje vrata bez potrebe za nošenjem mehaničkog ključa sa sobom.

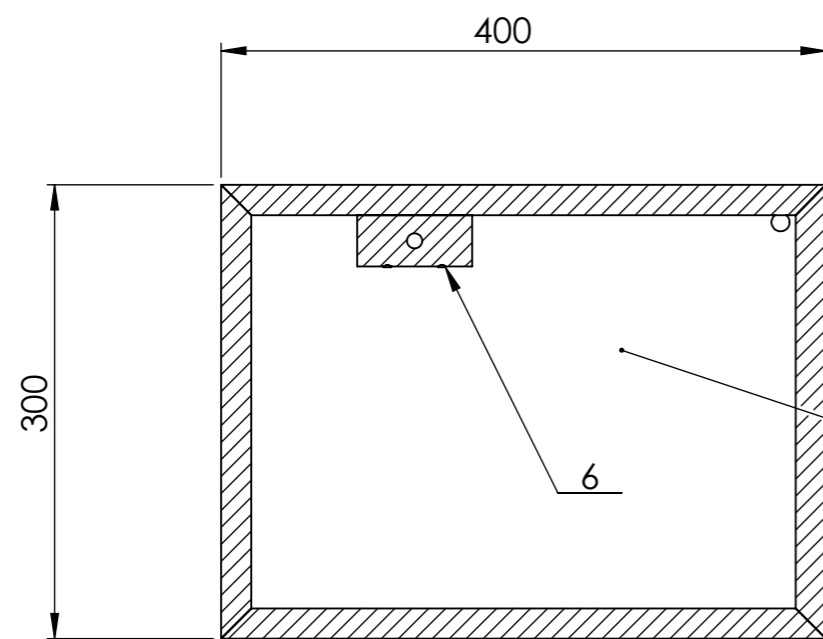
Tokom izrade ovog rada moguće je mnogo naučiti iz različitih područja mehatronike i općenito strojarstva, od programiranja i elektronike pa sve do konstruiranja fizičke izvedbe. Sva ova znanja vrlo su korisna za budućeg inženjera mehatronike i robotike.

LITERATURA

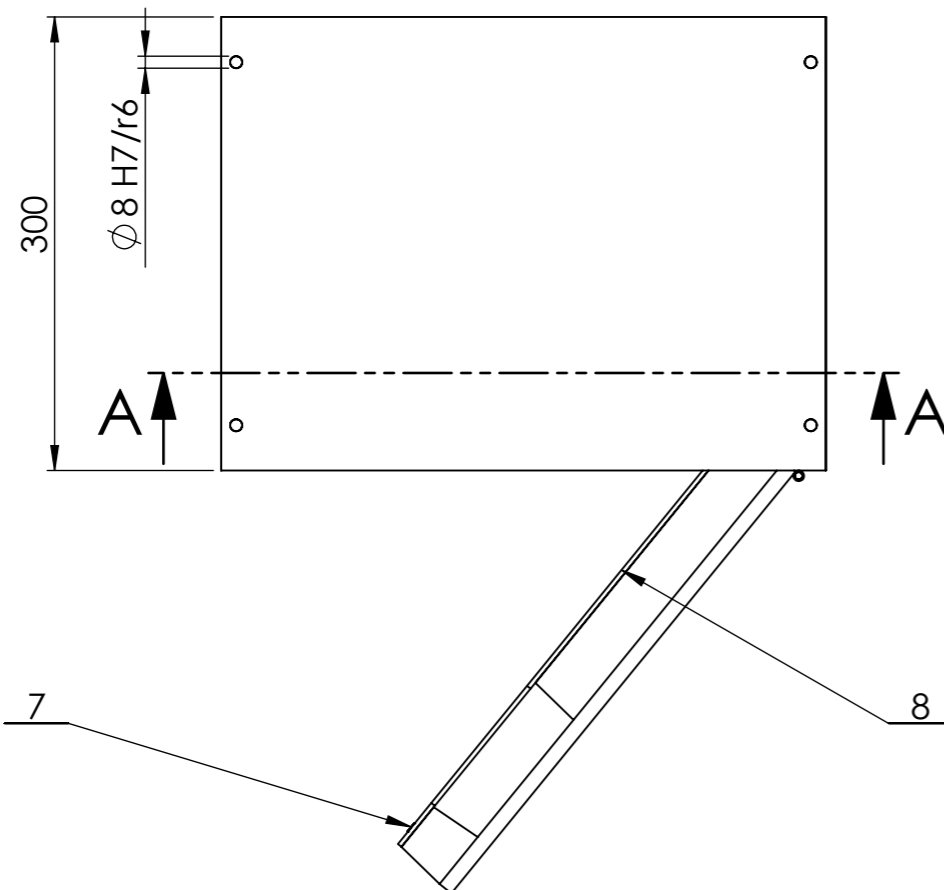
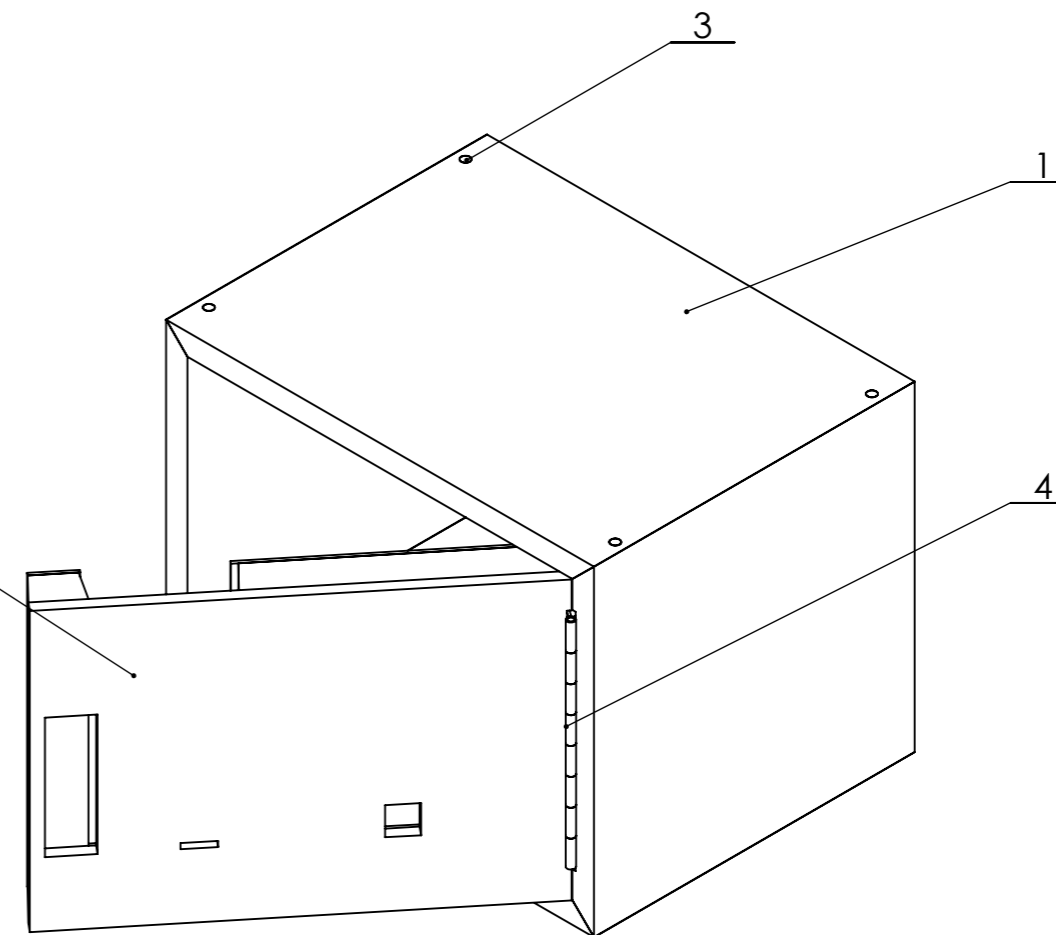
- [1] Elektronička brava secuENTRY, <https://www.burg.biz/international/p/electronic-door-locking-systems/entry-easy-5601-pincode/>
- [2] ECKey, <http://www.eckey.unis.hr/>
- [3] Arduino Mega 2560, <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [4] Arduino Mega 2560 Datasheet, https://www.alldatasheet.com/view.jsp?Searchword=Mega2560%20datasheet&gclid=Cj0KCQjw08aYBhDIARIsAA_gb0f_EGR0UFpCOFWFOIROJlhqMYj-tpZVSg6Bt0puyPamNKYaTjlstyYaAonNEALw_wcB
- [5] Tipkovnica, <https://e-radionica.com/hr/matrix-membranska-tipkovnica.html>
- [6] RFID RC522, <https://www.ad-electronic.hr/index.php/elektronika-i-elektrika%20-%202023/modul-rfid-rc522-13,56mhz-klju%C4%8D2274888-detalji>
- [7] Senzor otiska prsta, <https://e-radionica.com/hr/senzor-otiska-prsta.html>
- [8] Relej, <https://www.ad-electronic.hr/index.php/elektronika-i-elektrika%20-%202023/modul-relej-5v-10a-1kom8422486-detalji>
- [9] Elektromagnetski lokot, <https://e-radionica.com/hr/elektromagnetski-lokot-za-vrata-max-60kg.html>


PRILOZI

- I. CD-R disc
- II. Sklopni crtež



Presjek A-A



8	Poklopac za vrata	1				
7	Samourezni vijak 3,5x25	12	EN 14566			
6	Samourezni vijak 3,5x45	2	EN 14566			
5	Vrata	1				
4	Spojna šarka	1	DIN 7954 D			
3	Tipla drvena	8	DIN 68150			8x35
2	Zadnji poklopac	1				
1	Baza	1				
Poz.	Naziv dijela	Kom.	Crtež broj Norma	Materijal	Sirove dimenzije Proizvođač	Masa
Broj naziva - code		Datum	Ime i prezime	Potpis		
Projektirao		03.03.2022.	Marin Herak			
Razradio		03.03.2022.	Marin Herak			
Crtao		03.03.2022.	Marin Herak			
Pregledao						
Mentor						
ISO - tolerancije		Objekt:		Objekt broj:		
$\varnothing 8 H7$	+0,015 0			R. N. broj:		
$\varnothing 8 r6$	+0,028 +0,019	Napomena:				
$\varnothing 8 H7/r6$	-0,004 -0,028	Materijal:	Masa: 8 kg	ZAVRŠNI RAD		
		Naziv:		Pozicija:	Kopija	
		Sef		00	Format: A3	
		Mjerilo originala		Listova: 1		
		M 1:5		List: 1		
		Crtež broj: P-0000-00				