

# Automatsko otključavanje vrata s evidencijom ulazaka

---

**Leko, Tomislav**

**Undergraduate thesis / Završni rad**

**2016**

*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:235:681004>

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

*Download date / Datum preuzimanja:* **2024-12-28**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Tomislav Leko**

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentor:

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

Student:

Tomislav Leko

Zagreb, 2016.

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

Zahvaljujem se svojoj obitelji i zaručnici Martini na potpori i razumijevanju te svom mentoru dr.sc. Mladenu Crnekoviću koji mi je dao ideju za sam rad i pomogao savjetima u savladavanju prepreka za ostvarivanje iste u praksi.

Tomislav Leko



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



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

|  |        |
|--|--------|
| Sveučilište u Zagrebu<br>Fakultet strojarstva i brodogradnje |        |
| Datum  | Prilog |
| Klasa:   |        |
| Ur.broj:   |        |

## ZAVRŠNI ZADATAK

Student: **TOMISLAV LEKO**

Mat. br.: 0035176582

Naslov rada na hrvatskom jeziku: **AUTOMATSKO OTKLJUČAVANJE VRATA S EVIDENCIJOM ULAZAKA**

Naslov rada na engleskom jeziku: **AUTOMATIC DOOR UNLOCK WITH ENTRIES RECORDING**

Opis zadatka:

Klasično otključavanje i zaključavanje vrata mehaničkim ključem sve se više zamjenjuje elektroničkim bravama. Takva rješenja nude dodatne pogodnosti koje mehanički ključ nije imao, a jedna od njih je i automatska evidencija prolazaka.

Za vrata u nekoj ustanovi (npr. Fakultet strojarstva i brodogradnje) potrebno je projektirati automatski uređaj koji će registrirati dolazak osobe s ključem, otključati vrata i evidentirati prolazak ako se dogodi.

Tražena rješenja:

- odabrati i opisati rad potrebnih komponenata,
- projektirati elektronički sklop za detekciju elektroničkog ključa i otključavanje vrata,
- napisati program za evidenciju prolazaka,
- procijeniti vrijednost investicije.

Zadatak zadan:

25. studenog 2015.

Rok predaje rada:

1. rok: 25. veljače 2016.
2. rok (izvanredni): 20. lipnja 2016.
3. rok: 17. rujna 2016.

Predvideni datumi obrane:

1. rok: 29.2., 02. i 03.03. 2016.
2. rok (izvanredni): 30. 06. 2016.
3. rok: 19., 20. i 21. 09. 2016.

Zadatak zadao:

Prof.dr.sc. Mladen Crneković

Predsjednik Povjerenstva:

Prof. dr. sc. Zoran Kunica

## SADRŽAJ

|  |    |
|--|----|
| SADRŽAJ .....  | I  |
| POPIS SLIKA .....  | II |
| SAŽETAK.....   | IV |
| SUMMARY .....  | V  |
| 1. UVOD.....   | 1  |
| 1.1. Postojeća rješenja .....                              | 1  |
| 1.1.1. ECkey tehnologija.....                              | 1  |
| 1.1.2. SenseDoor tehnologija .....                         | 4  |
| 2. ODABIR OPREME.....                                      | 9  |
| 2.1. Komponente .....                                      | 9  |
| 2.2. Arduino UNO tiskana pločica.....                      | 9  |
| 2.3. Napajanje .....                                       | 14 |
| 2.4. RTC ( <i>Real-Time Clock</i> ) i EEPROM memorija..... | 15 |
| 2.5. Relej s optocouplerom .....                           | 17 |
| 2.6. RFID čitač .....                                      | 18 |
| 2.7. Dodatni materijali .....                              | 21 |
| 3. ELEKTRONIČKA BRAVA .....                                | 22 |
| 3.1. Simulacijski dio.....                                 | 22 |
| 3.2. Eksperimentalni dio .....                             | 23 |
| 3.3. Postupak izrade sklopa.....                           | 23 |
| 3.4. Alternativni postupak izrade .....                    | 27 |
| 4. PROGRAMIRANJE.....                                      | 31 |
| 4.1. Logika programiranja.....                             | 33 |
| 4.2. Kontrola pristupa.....                                | 35 |
| 4.3. Programski kod .....                                  | 35 |
| 5. ZAKLJUČAK.....  | 44 |
| LITERATURA.....  | 45 |
| PRILOZI.....   | 46 |

## POPIS SLIKA

|           |   |    |
|-----------|---|----|
| Slika 1.  | Prikaz upotrebe virtualne tipkovnice uz pomoć mobitela .....                                  | 2  |
| Slika 2.  | Prikaz upotrebe virtualne tipkovnice uz pomoć mobitela .....                                  | 3  |
| Slika 3.  | Procesni dijagram ECKey sustava na bežično upravljanje .....                                  | 3  |
| Slika 4.  | Čitač kartica.....  | 5  |
| Slika 5.  | Čitač kartice u primjeni (vanjska strana prostorije) .....                                    | 5  |
| Slika 6.  | Držač kartica.....  | 6  |
| Slika 7.  | Držač kartice u primjeni (unutarnja strana prostorije) .....                                  | 6  |
| Slika 8.  | Kontroler zone (GZC) .....  | 7  |
| Slika 9.  | Elektroprihvatanik.....   | 7  |
| Slika 10. | SenseDoor brava za zaključavanje .....  | 7  |
| Slika 11. | Arduino UNO tiskana pločica .....   | 10 |
| Slika 12. | Tehničke specifikacije Arduino UNO pločice .....  | 12 |
| Slika 13. | Shematski prikaz Arduino UNO pločice (Altium Designer) .....                                  | 13 |
| Slika 14. | DC-DC modul napajanja .....   | 14 |
| Slika 15. | Shematski prikaz DC-DC modula napajanja.....  | 15 |
| Slika 16. | RTC i EEPROM.....   | 16 |
| Slika 17. | Shematski prikaz RTC i EEPROM .....   | 16 |
| Slika 18. | Relej s optocouplerom.....  | 17 |
| Slika 19. | Shematski prikaz modula releja s optocouplerom.....   | 18 |
| Slika 20. | RFID čitač s privjeskom i karticom .....  | 19 |
| Slika 21. | Shematski prikaz RFID čitača .....  | 20 |
| Slika 22. | Taster za pločicu.....  | 21 |
| Slika 23. | RGB LED dioda .....   | 21 |
| Slika 24. | Električna shema spoja svih modula .....  | 22 |
| Slika 25. | Električna shema spoja svih modula .....  | 23 |
| Slika 26. | Prototipna pločica tijekom postupka lemljenja (stražnja strana).....                          | 24 |
| Slika 27. | Prototipna pločica nakon postupka lemljenja (prednja strana) .....                            | 24 |
| Slika 28. | Prikaz kućišta s utorom za karticu.....   | 25 |
| Slika 29. | Prikaz kućišta sa otvorenim poklopcem.....  | 26 |
| Slika 30. | Prikaz sklopa za demonstraciju rada .....   | 27 |
| Slika 31. | 2D prikaz PCB-a u Altium Designer-u .....   | 28 |
| Slika 32. | 3D prikaz PCB-a u Altium Designer-u .....   | 28 |
| Slika 33. | Prikaz tiskane pločice ( <i>Final Artwork Print</i> ) .....                                   | 29 |
| Slika 34. | Prikaz prostornog rasporeda komponenata u Altium Designer-u ( <i>Assembly Drawing</i> ) ..... | 29 |
| Slika 35. | Arduino IDE sučelje.....  | 32 |
| Slika 36. | Izbornik s prečacima .....  | 33 |
| Slika 37. | Logika za programiranje .....   | 34 |
| Slika 38. | Definiranje varijabli – RFID .....  | 35 |
| Slika 39. | Definiranje varijabli – digitalni izvodi .....  | 35 |
| Slika 40. | Definiranje varijabli – RTC.....  | 36 |
| Slika 41. | Definiranje varijabli – EEPROM .....  | 36 |
| Slika 42. | RTC funkcije .....  | 38 |
| Slika 43. | EEPROM funkcije.....  | 40 |
| Slika 44. | Glavni dio programa ( <i>setup</i> ).....   | 41 |

Slika 45. Glavni dio programa (*loop*) ..... 43



## SAŽETAK

U sklopu završnog rada u početku je dan kratak uvod o značaju upotrebe beskontaktnog otvaranja vrata i osvrt na slične tehnologije koje se danas primjenjuju. U drugom dijelu je dan popis i opis opreme koja se koristila za realizaciju samog zadatka, a to je otvaranje elektroničke brave i evidencija prolazaka. 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 izvađen nakon provedbe cijelog završnog rada, od teorijskog ka praktičnom dijelu.

## **SUMMARY**

This undergraduate thesis provides a short introduction to the use of contactless 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 and logging. 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

U današnje vrijeme klasično otključavanje i zaključavanje vrata mehaničkim ključem sve se više zamjenjuje elektroničkim bravama. Takva rješenja nude dodatne pogodnosti koje nadilaze sposobnosti mehaničkog ključa, kao što je jednostavniji i brži ulazak u prostoriju, automatska registracija prolazaka itd. Užurbani ritam kojeg nameće svakodnevnica u čovjekovom životu može se znatno olakšati korištenjem modernijih i jednostavnijih pristupa nekom objektu, uredu i sl.

### 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. ECKey tehnologija

Jedno od njih je tzv. ECKey tehnologija čija kampanja se vodi sloganom: „Pretvorite svoj mobilni u ključ...“. Ovakva tehnologija na tržištu omogućava mobilnim Bluetooth uređajima (poput mobitela ili PDA uređaja) da služe kao visoko sigurnosni ključ za otključavanje/zaključavanje automobila, ulaznih vrata, dvorišnih vrata, garažnih vrata, parking barijera, ureda, kućnih sigurnosnih sustava, liftova, sportskih klubova, hotela, apartmana, alarmnih uređaja ili sustava za kontrolu pristupa. ECKey uređaj ima mogućnost identifikacije mobitela na siguran način, koristeći pri tome jedinstvenu Bluetooth adresu i sigurnu 128-bitnu enkripciju. Uz pomoć automatske „hands-free“ kontrole pristupa mogu se uporabom mobitela ili PDA s uključenim Bluetooth-om automatski zaključavati i otključavati bilo koja vrata s ugrađenim ECKey uređajem. Nova ECKey tehnologija jednostavno se integrira s postojećim vratima. Potrebna je jednokratna registracija nakon koje će određeni mobilni uređaj automatski zaključati vrata prilikom odlaska, i automatski otključati vrata prilikom povratka. Pri tome nije potrebno pritiskati nikakve gumbe, niti tražiti ključeve, ili čak izvaditi mobilni iz džepa ili torbe. 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 (<http://www.eckey.unis.hr/>).

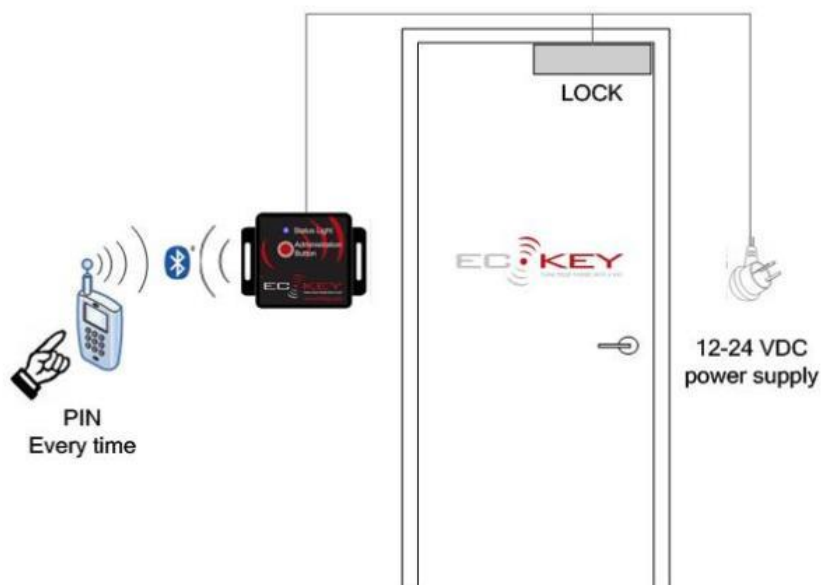
ECkey uređaji su proizvedeni u različitim modelima i s različitom primjenom: mogu biti samostalni ili integrirani s postojećim sustavima, mogu imati offline ili online kontrolu pristupa.

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 [Slika 1]. Može se instalirati s unutarnje strane zbog zaštite od nevremana i vandalizma.



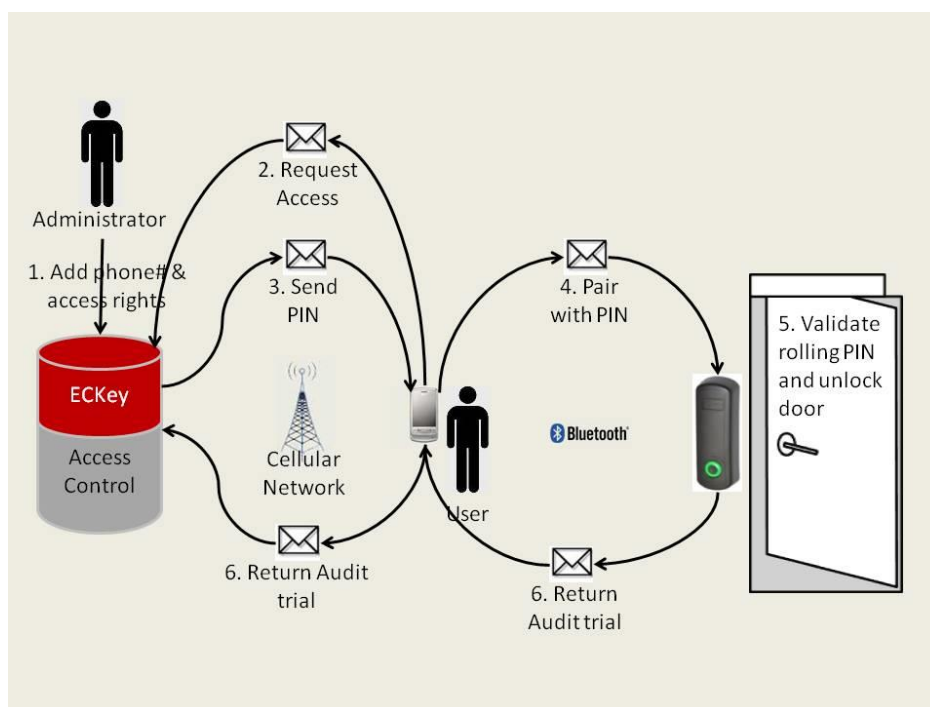
**Slika 1. Prikaz upotrebe virtualne tipkovnice uz pomoć mobitela**

Procesni dijagram ovog sustava prikazan je na slici [Slika 2]. Korisnik definira PIN od 1 – 8 znamenki, podešivi raspon može biti od jednog metra do deset metara, a napajanje je od 12 – 24 VDC s potrošnjom 20 mA pri 12 VDC. U ovom slučaju za mobitel ili PC nije potreban nikakav dodatan softver, dok se kao podrška može koristiti online pristup bazi znanja.



Slika 2. Prikaz upotrebe virtualne tipkovnice uz pomoć mobitela

Procesni dijagram druge vrste sustava u centralnom upravljanju udaljenim lokacijama prikazan je na slici [Slika 3]. Prikazani sustav se sastoji od izdavanja rotirajućih PIN-ova putem SMS-a na mobitel prema određenom rasporedu ili *ad-hoc* pristupu.



Slika 3. Procesni dijagram ECkey sustava na bežično upravljanje

Neke od prednosti ECKey tehnologije su sljedeće:

- Nedostatak vidljivih elemenata koji bi bili predmet napada provalnika
- Nedostatak kartica ili ključeva koje treba distribuirati i vraćati
- Sposobnost integracije u postojeći sustav za kontrolu pristupa ili kao samostalni sustav
- Podesivi raspon od pola metra do deset metara
- Mogućnost identifikacije za vrijeme vožnje bez zaustavljanja
- Rad bez potrebe instaliranja ikakvog softvera na mobitel

### ***1.1.2. SenseDoor tehnologija***

Jedan od sustava koji se također nudi na tržištu kao rješenje za automatsko upravljanje i regulaciju otključavanja i zaključavanja brave dolazi od talijanske tvrtke GFP LAB sa sjedištem u Riminiju. Tehnologija koja se koristi naziva se „SenseDoor“. Talijanska tvrtka specijalizirana u hotelskoj automatizaciji je ovu tehnologiju primijenila kao kontrolu sustava za pristup dizajnirana da zaštiti bravu od vrata. SenseDoor tehnologija ne zahtjeva nikakve pripreme ili zidarske radove jer se jednostavno instalira na okvir od vrata. Može se napajati pomoću baterija i može komunicirati bežično sa drugim uređajima i sensorima. Svestranost SenseDoor sustava omogućava da se adaptira svim mogućim scenarijima, od jednostavnog ulaska u prostoriju, aktivacije same sobe i termoregulaciju, do potpune integracije sa Giano sustavom (integriranim hotelskim sustavom upravljanja). Postoje razni proizvođači diljem svijeta koji nude slična rješenja koja se tiču otključavanja/zaključavanja objekata i upravljanja kako za primjenu u industriji, ugostiteljstvu, privatnom sektoru, poslovnom sektoru itd. Navedene su samo neke od mogućih varijanti korištenja tih sustava kako bi se mogle usporediti sa samim rješenjem zadatka ovog završnog rada. Neke od osnovnih komponenti koje čine SenseDoor sustav opisane su u nastavku.

Čitač kartica (ZDR) prikazan na slici [Slika 4] je eksterni RFID čitač od SenseDoor tehnologije. On je predviđen da bude pozicioniran izvan prostorije na okviru vrata tako da kontrolira pristup u prostoriju i prikazuje status prostorije (okupirana, ne smetaj, alarm itd.). Također može biti korišten i u zajedničkim područjima.



**Slika 4. Čitač kartica**

U primjeni već ugrađeni čitač kartice može se vidjeti na slici [Slika 5].



**Slika 5. Čitač kartice u primjeni (vanjska strana prostorije)**

Držać kartica [Slika 6] predstavlja unutarnji RFID čitač koji je pozicioniran unutar sobe, na okvir vrata. Prilikom umetanja ispravne kartice omogućava usluge sobe, svjetla i ventilaciju (klimatiziranje). Stanje privatnosti i neuznemiravanja (engl. *do not disturb*) aktivira se pritiskom na tipku. Ista tipka može se koristiti i za pozivanje usluge čistačice.



**Slika 6. Držač kartica**

U primjeni već ugrađeni držač kartice može se vidjeti na slici [Slika 7].



**Slika 7. Držač kartice u primjeni (unutarnja strana prostorije)**

Također je potrebno imati sam čitač kartice [Slika 4] koji predstavlja eksterni RFID čitač od SenseDoor tehnologije. On je pozicioniran s vanjske strane prostorije, na okviru vrata te kontrolira pristup prostoriji i prikazuje status sobe (zauzeto, ne smetaj, alarm, itd...).

Informacije sa svih sobnih uređaja koje su spojene na njega sakuplja kontrolor zone (GZC) prikazan na slici [Slika 8]. U generalnom slučaju ovisno o vrsti zgrade uobičajeno je postavljen na jednom od katova hotela (unutar električnog panela ili unutar lažnog stropa).





**Slika 8. Kontroler zone (GZC)**

Električno-mehanička komponenta koja fizički otključava vrata naziva se elektroprihvatnik [Slika 9] i ona je ugrađena u dovratnik.



**Slika 9. Elektroprihvatnik**

SenseDoor brava za zaključavanje [Slika 10] je osnovni modul od SenseDoor tehnologije koji pogoni i spaja oba čitača i gdje se nalaze svi električni spojevi. Ona je ugrađena u dovratniku samih vrata.



**Slika 10. SenseDoor brava za zaključavanje**

Postoje još razne mogućnosti implementacije dodatnih senzora i uređaja na sam sustav s ciljem poboljšanja energetske učinkovitosti, sigurnosnih mjera, luksuza samoga korisnika i sl. ali sve osnovne komponente sustava su navedene ranije.

## 2. ODABIR OPREME

### 2.1. Komponente

Prvi korak pri samoj realizaciji nakon pomno osmišljene ideje jest kako ostvariti tu ideju. Da bi se taj korak uspješno izvršio potrebno je odabrati same komponente koje će se koristiti za obavljanje predviđene zadaće.

Korišteni su već postojeći proizvodi na tržištu koji integrirani zajedno na pravilan način mogu ostvariti zadaću, a navedeni su kako slijedi:

- Arduino UNO R3 (tiskana pločica)
- 3A DC-DC Converter 1pc modul (napajanje visoke efikasnosti)
- MFRC-522 RFID (RFID čitač s tagovima)
- Tiny RTC I2C DS1307 AT24C32 (memorija i RTC modul za Arduino)
- 5V 10A 1 Channel Relay Modul (releji s optocouplerom)

Navedeni uređaji nalaze se zasebno svaki na svojoj tiskanoj pločici (to nazivamo modulom) sa svim potrebnim komponentama koje zajedno čine rad sklopa.

Uz gotove module potrebne su dodatne fizikalne komponente i softver koji su navedeni ispod:

- RGB LED (dioda)
- 100 PCs Tactile Push Button Switch (botuni)
- Arduino IDE sučelje (sučelje za programiranje Arduina)

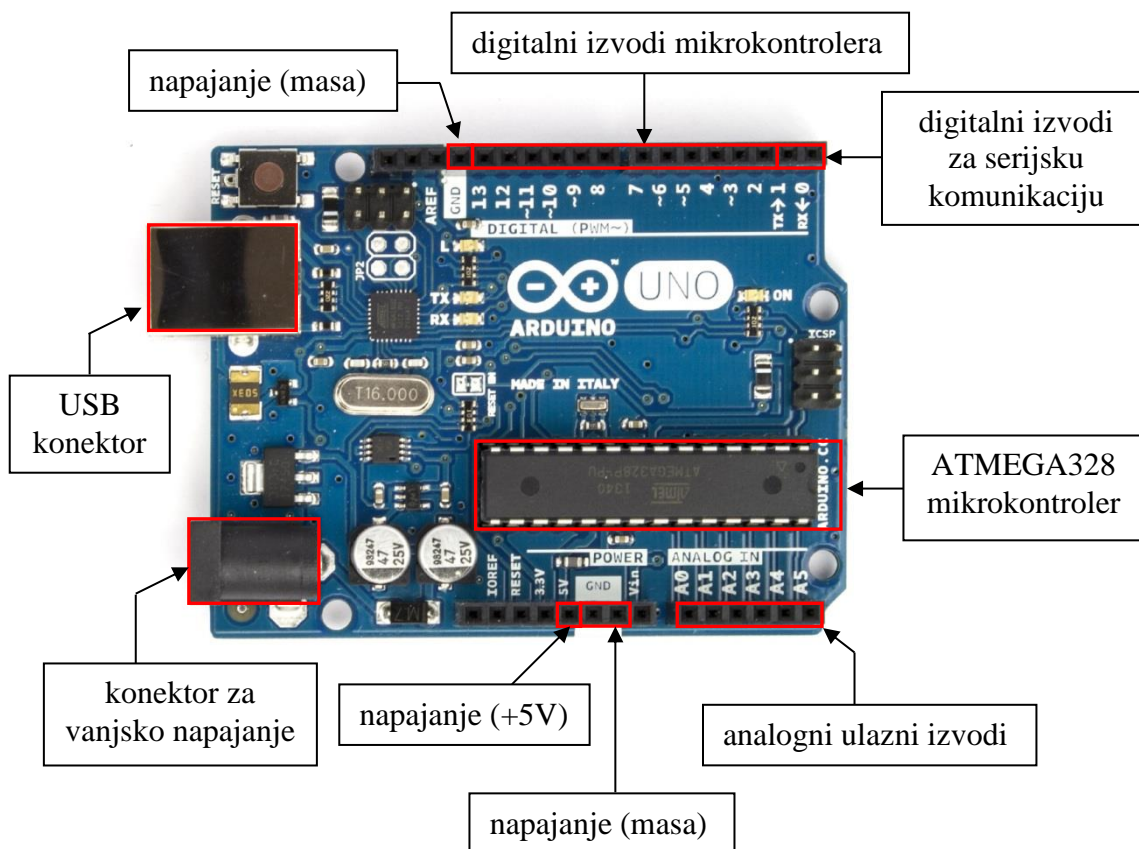
### 2.2. Arduino UNO tiskana pločica

Arduino projekt započeo je 2005. godine na sveučilištu u Italiji (Ivery) kao platforma za učenje programiranja i korištenje mikrokontrolera. Mikrokontroler korišten na ovoj izuzetno popularnoj Arduino platformi je ATmega328 koji proizvodi tvrtka ATMEL. Pločica Arduino sastoji se od Atmel 8-bitnih, 16-bitnih i 32-bitnih AVR mikrokontrolera s komplementarnim komponentama koje olakšavaju programiranje i ugradnju u druge strujne krugove i sustave.

Od raznoraznih Arduino pločica (engl. *boards*) odabrana je Arduino UNO pločica [Slika 11] s obzirom da ona predstavlja jedan od boljih izbora pločica za početak rada sa elektronikom i samim kodiranjima. Također sadrži najopsežniju i najčešće korištenu dokumentaciju od cijele Arduino & Genuino obitelji, te je dosta robusna. Kao što je ranije navedeno UNO je pločica

bazirana na ATmega328P platformi. Sastoji se od 14 digitalnih ulazno/izlaznih pinova (od kojih se 6 može koristiti kao PWM izlaz), 6 analognih ulaza, kristalnog oscilatora od 16 MHz, USB priključka, priključka za napajanje, ICSP *header*-a i tipke za reset. Za početak rada može se jednostavno spojiti na računalo pomoću USB priključka, napajati pomoću *AC-to-DC* adaptera ili najobičnije baterije. Sama robusnost izvedbe omogućava jednostavnije i sigurnije korištenje bez opasnosti, raznorazna isprobavanja i ispitivanja, a u slučaju da nešto pođe po zlu može se jednostavno zamijeniti sam čip i pokušati ponovno.

Oznaka „UNO“ na samoj pločici znači „jedan“ na talijanskom i odabrana je kao oznaka za najavu novog proizvoda za Arduino Softver (IDE) 1.0. Pločica UNO i verzija 1.0 od Arduino Softvera (IDE) bile su referentne verzije Arduina i do sada su već razvijene novije verzije. Arduino UNO je također prva u seriji USB Arduino pločica kao i referentni model za Arduino platformu.



Slika 11. Arduino UNO tiskana pločica

Odabran je Arduino s obzirom da je on općenito jedan od najpoznatijih alata koji se u svijetu najviše koristi upravo za učenje i početak rada s mikrokontrolerima.

Arduino je, kao platforma za učenje programiranja i korištenja mikrokontrolera, nastao prije desetak godina. To je tzv. „open-source“ platforma za kreiranje elektroničkih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje. Arduino je namijenjen umjetnicima, dizajnerima, hobistima, elektroničarima i svima koji su zainteresirani za stvaranje interaktivnih objekata ili okruženja.

Cijela Arduino platforma napravljena je i objavljena od strane proizvođača kao otvoreni sustav – to znači da su sve sheme i izvorni kod programa za programiranje Arduina besplatno dostupni svima za preuzimanje i modificiranje sa službene stranice Arduino platforme ([www.arduino.cc](http://www.arduino.cc)).

Glavni dio Arduino UNO pločice je ATmega328 mikrokontroler. Sklopovlje koje se nalazi oko njega služi kako bi se mikrokontroler mogao pokrenuti i kako bismo u njega mogli ubaciti program koji se piše na računalu. Komunikacija Arduino UNO pločice i računala odvija se preko USB priključka.

Za programiranje mikrokontrolera može se koristiti Arduino IDE programsko sučelje. Program se može besplatno preuzeti na web-adresi proizvođača.

Za prikaz spajanja Arduino UNO pločice sa ostalim modulima tijekom projektiranja elektroničke brave korišten je besplatni program pod nazivom Fritzing ([www.fritzing.org](http://www.fritzing.org)). To može biti vrlo koristan alat za dokumentiranje projekata s mikrokontrolerima.

Izvodi mikrokontrolera su na Arduino UNO pločici [Slika 11] spojeni na crne konektore na samom rubu pločice. Svaki izvod ima svoje jedinstveno ime koje je napisano kraj njega. Tako se na konektoru s gornje strane nalaze izvodi 0, 1, 2, 3 itd. do 13. Sa donje strane Arduino UNO pločice dodatno se nalaze izvodi koji se nazivaju A0, A1, A2 itd. do A5. Dodatno u donjem dijelu pločice postoje i VCC (+5V) i GND izvodi. Osim ovih izvoda na pločici su označeni i drugi izvodi.

Izvodi mikrokontrolera služe za povezivanje mikrokontrolera s elektroničkim elementima, sklopovima i uređajima koji se nalaze izvan Arduino pločice. Mogu se koristiti za upravljanje svjetlećim diodama, motorima itd. ili za očitavanje stanja s različitih senzora (tipkala,

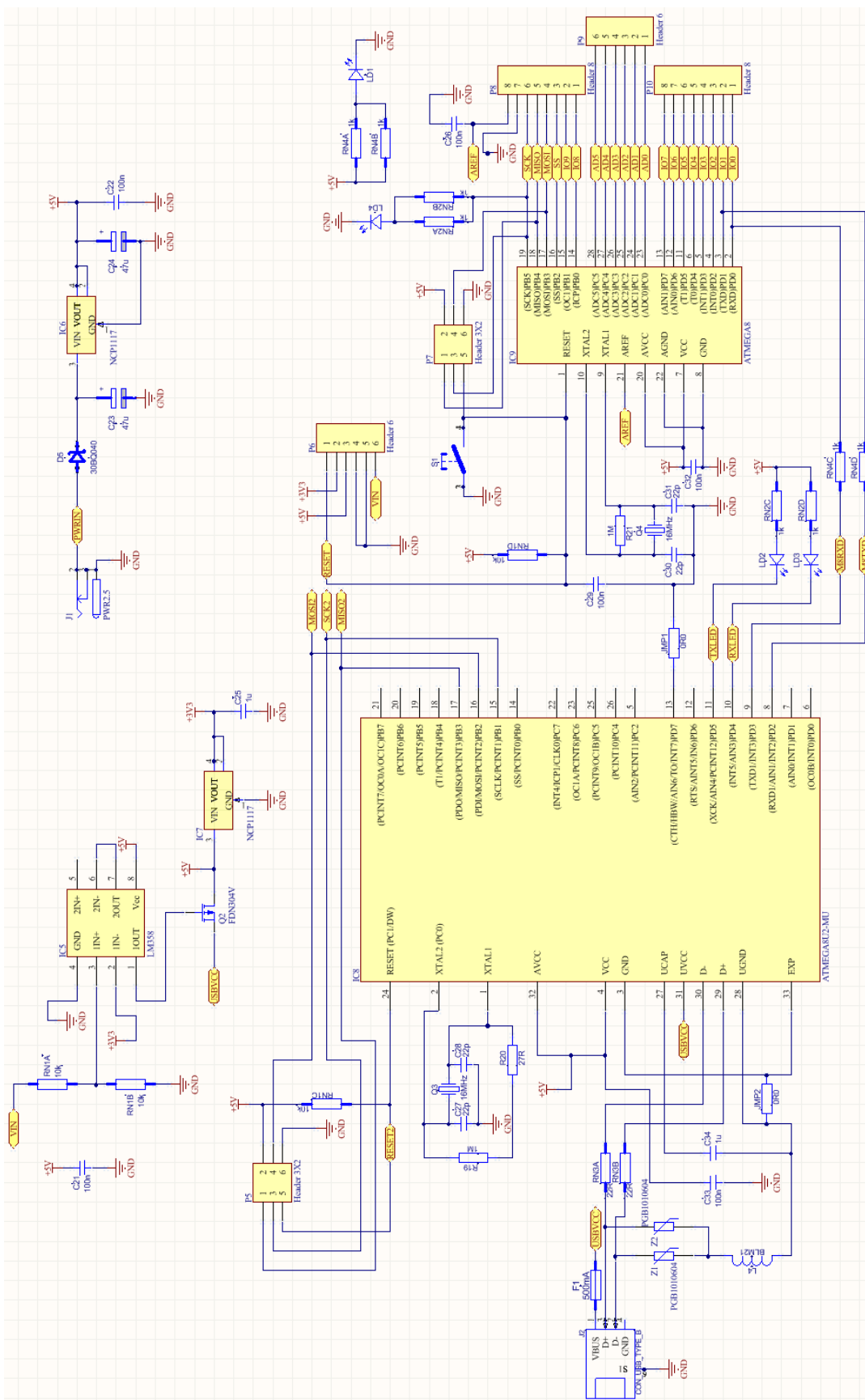
sklopke, senzori temperature itd.). Tako kažemo da izvodi mogu imati dva načina rada – izlazni i ulazni.

Tehničke specifikacije od same pločice preuzete su sa stranice proizvođača [Slika 12].

|                             |  |
|-----------------------------|--|
| Microcontroller             | ATmega328P   |
| Operating Voltage           | 5V   |
| Input Voltage (recommended) | 7-12V  |
| Input Voltage (limit)       | 6-20V  |
| Digital I/O Pins            | 14 (of which 6 provide PWM output)                       |
| PWM Digital I/O Pins        | 6  |
| Analog Input Pins           | 6  |
| DC Current per I/O Pin      | 20 mA  |
| DC Current for 3.3V Pin     | 50 mA  |
| Flash Memory                | 32 KB (ATmega328P)<br>of which 0.5 KB used by bootloader |
| SRAM                        | 2 KB (ATmega328P)  |
| EEPROM                      | 1 KB (ATmega328P)  |
| Clock Speed                 | 16 MHz   |
| LED_BUILTIN                 | 13   |
| Length                      | 68.6 mm  |
| Width                       | 53.4 mm  |
| Weight                      | 25 g   |

**Slika 12. Tehničke specifikacije Arduino UNO pločice**

Shematski prikaz je zasebno nacrtan u programu Altium Designer u svrhu izrade zasebne tiskane pločice sa svim komponentama sa svih modula u jednoj cjelini [Slika 13].



Slika 13. Shematski prikaz Arduino UNO pločice (Altium Designer)

### 2.3. Napajanje

Kao napajanje samog sklopa može se koristiti USB priključak koji tada može spojiti Arduino UNO pločicu sa računalom. Također se može staviti zasebna baterija od 12 V ili spojiti eksterno napajanje. DC-DC modul odabran je kao *switching* napajanje visoke efikasnosti, što nam omogućava korištenje bilo kojeg ulaznog napona (6-24V). Takva varijanta radila bi bez zagrijavanja, za razliku od tipičnih 7805 neefikasnih linearnih regulatora.

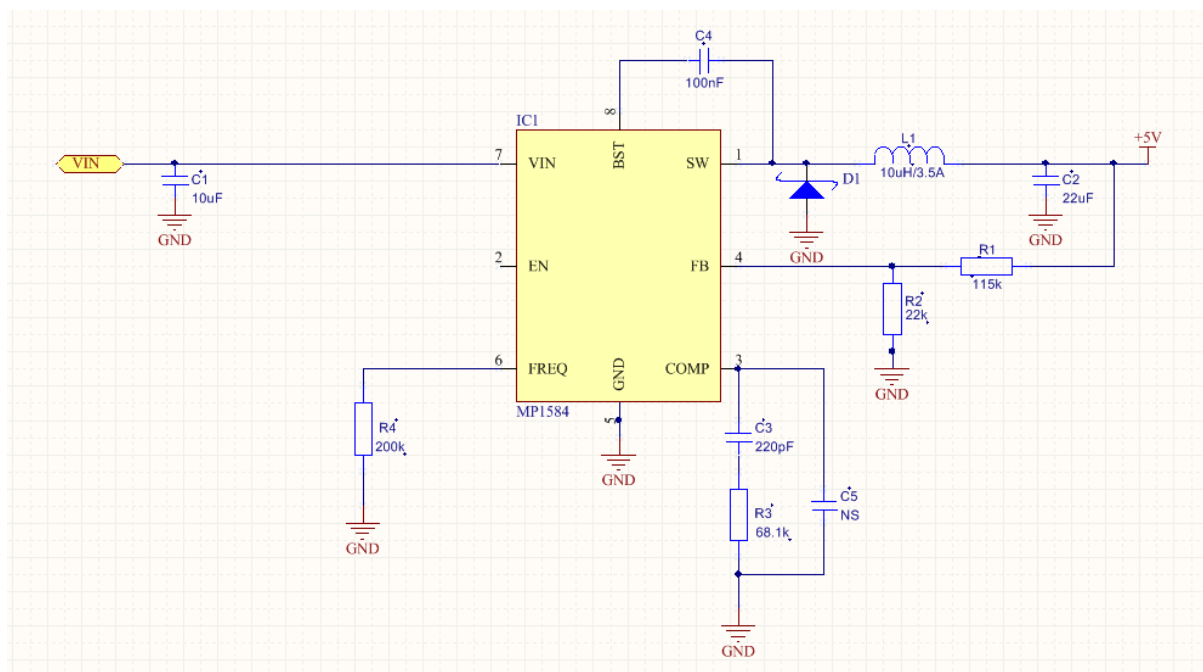
DC-DC modul [Slika 14] je u konačnici izostavljen iz samog sklopa radi uštede prostora, kompaktnosti izvedbe i smanjenja cijene ali svakako se može koristiti kako će kasnije biti prikazano na slici sveukupne sheme sklopa.



Slika 14. DC-DC modul napajanja

Također u svrhu izrade zasebne pločice modul je uz pomoć sheme proizvođača nacrtan u programu Altium Designer [Slika 15].





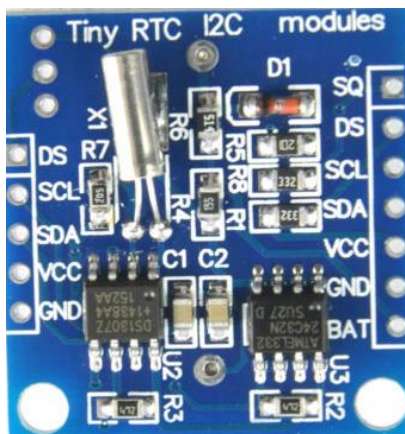
Slika 15. Shematski prikaz DC-DC modula napajanja

#### 2.4. RTC (*Real-Time Clock*) i EEPROM memorija

RTC (*Real Time Clock*) je tzv. računalni sat (najčešće u formi integriranog kruga) koji prati trenutno vrijeme u stvarnosti, stoga je izabran radi bilježenja kontrole pristupa u formatu datum/godina/vrijeme.

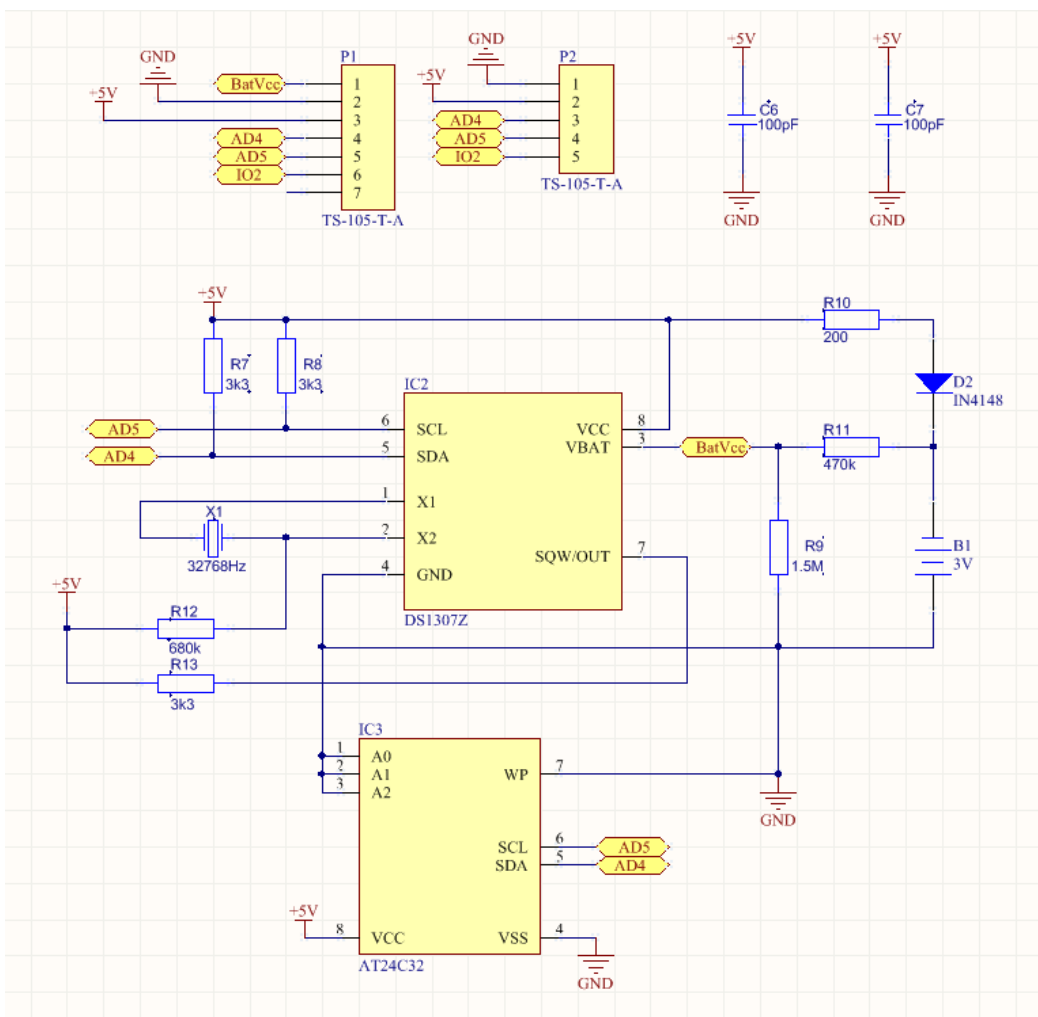
EEPROM (*Electrically Erasable Programmable Read-Only Memory*) je vrsta ispisne memorije koja se može brisati i ponovno programirati električnim signalom (brže u odnosu na ultraljubičasto zračenje). Prednost ROM memorije je u tome što čuva podatke u slučaju nestanka struje. Za naš slučaj električne brave, u EEPROM se može zapisati posljednjih N ulazaka, kartica koja je korištena za ulaz, te kartice koje će imati omogućen pristup.

Izabran je modul kompatibilan sa Arduino UNO pločicom koji sadrži i RTC i EEPROM na jednoj pločici i prikazan je na slici [Slika 16].



Slika 16. RTC i EEPROM

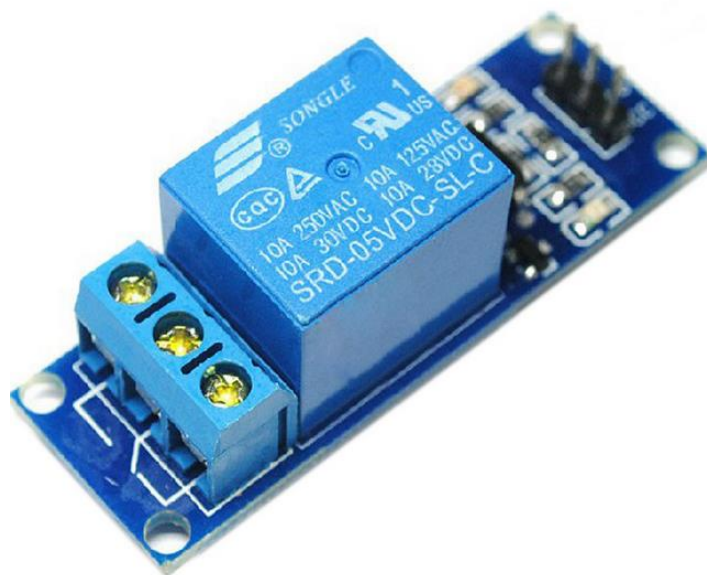
Također u svrhu izrade zasebne pločice modul je uz pomoć sheme proizvođača nacrtan u programu Altium Designer [Slika 17].



Slika 17. Shematski prikaz RTC i EEPROM

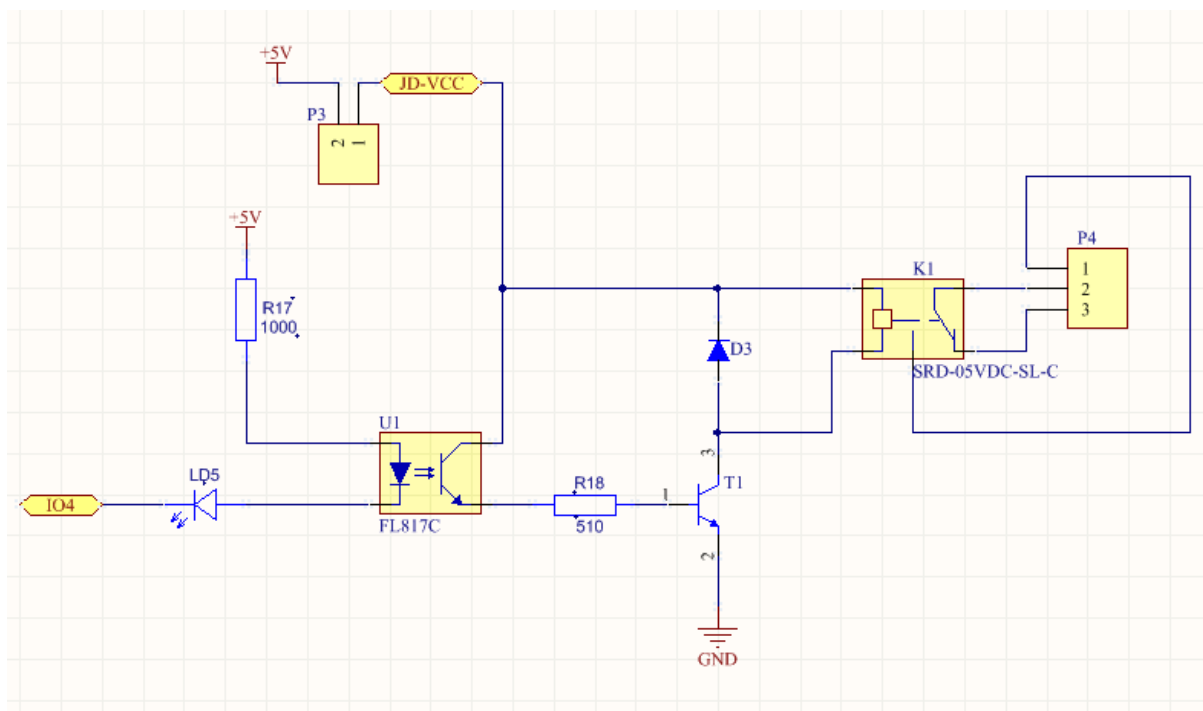
## 2.5. Relej s optocouplerom

Navedeni modul služiti će nam za aktiviranje solenoida vrata na sljedećem principu – uključivanjem releja pokrenuti će se solenoid (brava) što će omogućiti ulaz korisniku u prostoriju. Izabran je relej s optocouplerom da se potpuno izolira mikrokontrolerska strana od „visokonaponske“ gdje na releju vladaju prijelazne pojave koje mogu uništiti mikrokontroler. Cijeli modul prikazan je na slici [Slika 18].



Slika 18. Relej s optocouplerom

Također u svrhu izrade zasebne pločice modul je uz pomoć sheme proizvođača nacrtan u programu Altium Designer [Slika 19].



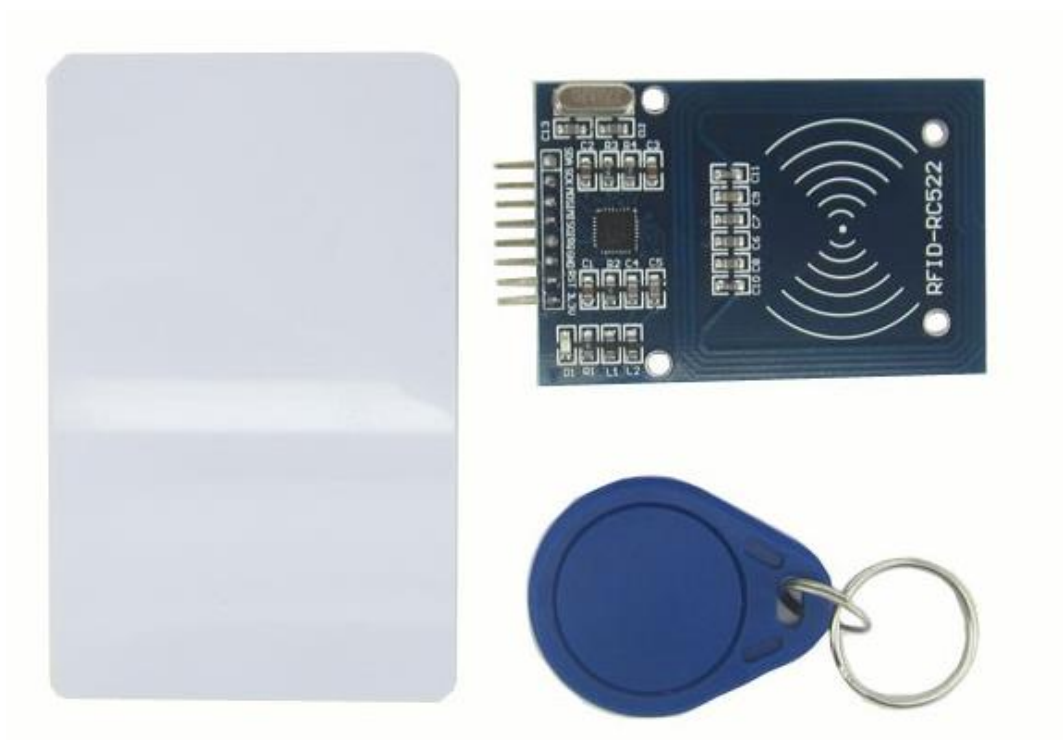
Slika 19. Shematski prikaz modula releja s optocouplerom

## 2.6. RFID čitač

RFID (engl. *Radio Frequency Identification*) je tehnologija koja koristi tehniku frekvencijskih radiovalova za izmjenu podataka između čitača (*reader*) i uređaja koji se naziva transmiter (tag).

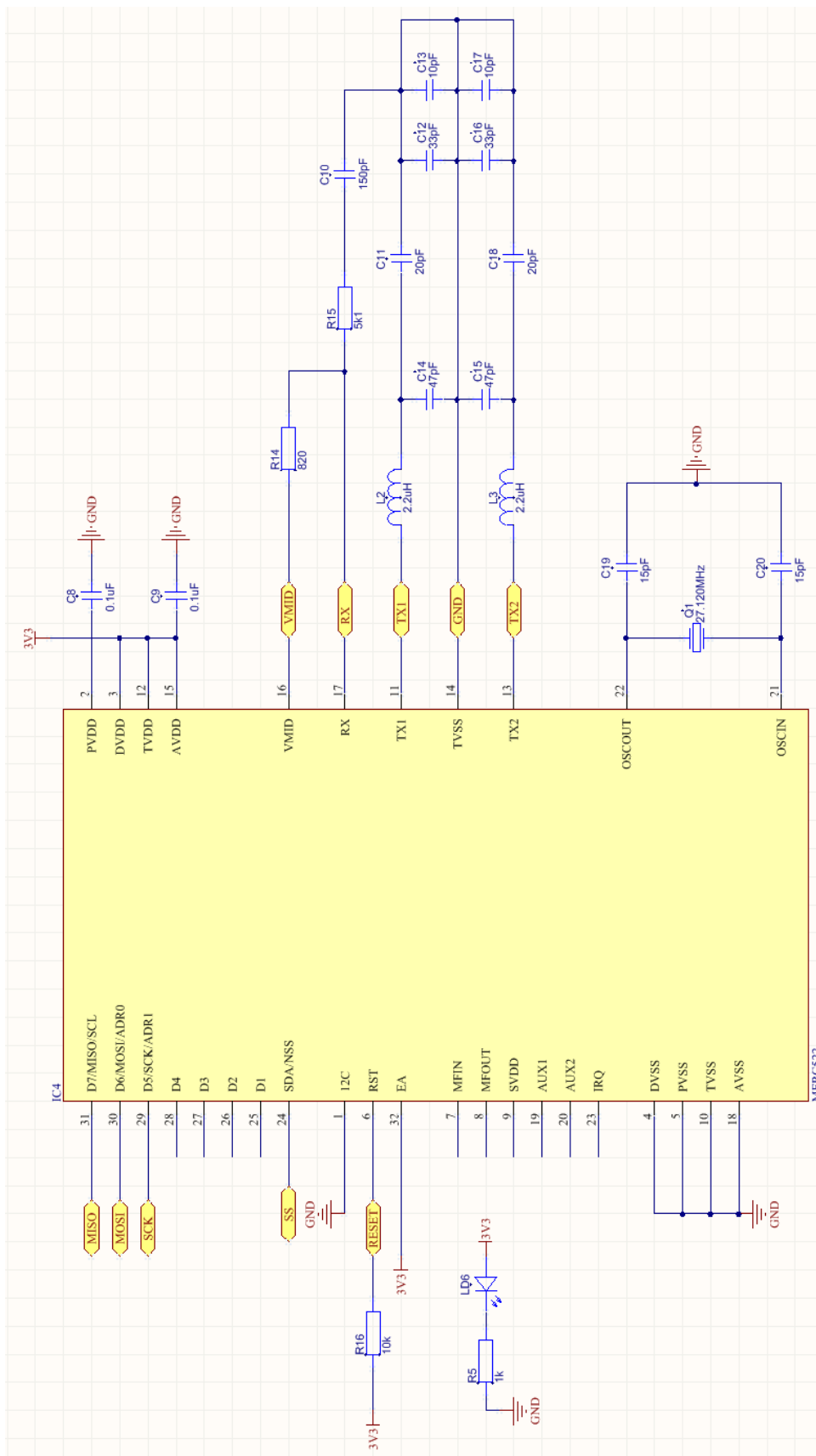
Razlog za izbor ovog čitača je mogućnost i predviđenost komunikacije sa Arduino UNO pločicom sa detaljnim instrukcijama spajanja *portova*, relativno male dimenzije te niska cijena. Radna frekvencija uređaja je 13.56 MHz.

Pomoću RFID čitača omogućiti će se čitanje kartica (ili narukvica) sa RFID kodom. Cijeli paket prikazan je na slici [Slika 20].



**Slika 20. RFID čitač s privjeskom i karticom**

Također u svrhu izrade zasebne pločice modul je uz pomoć sheme proizvođača nacrtan u programu Altium Designer [Slika 21].



Slika 21. Shematski prikaz RFID čitača

## 2.7. Dodatni materijali

### TASTER

Ukoliko se odlučimo na zasebne elektromehaničke komponente koje fizički možemo aktivirati pritiskom na njih onda u tu svrhu možemo upotrijebiti tastere za pločicu [Slika 22]. Njima ćemo omogućiti pamćenje kartice koju želimo dodati za otvaranje vrata (resetiranje i pohrana podataka).



Slika 22. Taster za pločicu

U ovom slučaju način pohrane riješen je u programu softverski tako da na samom sklopu nisu korišteni zasebni tasteri.

### RGB LED

Odabrana komponenta [Slika 23] signalizirati će aktivnost uređaja:

- plava (uključeno)
- crvena (odbijen pristup)
- zelena (omogućen pristup)



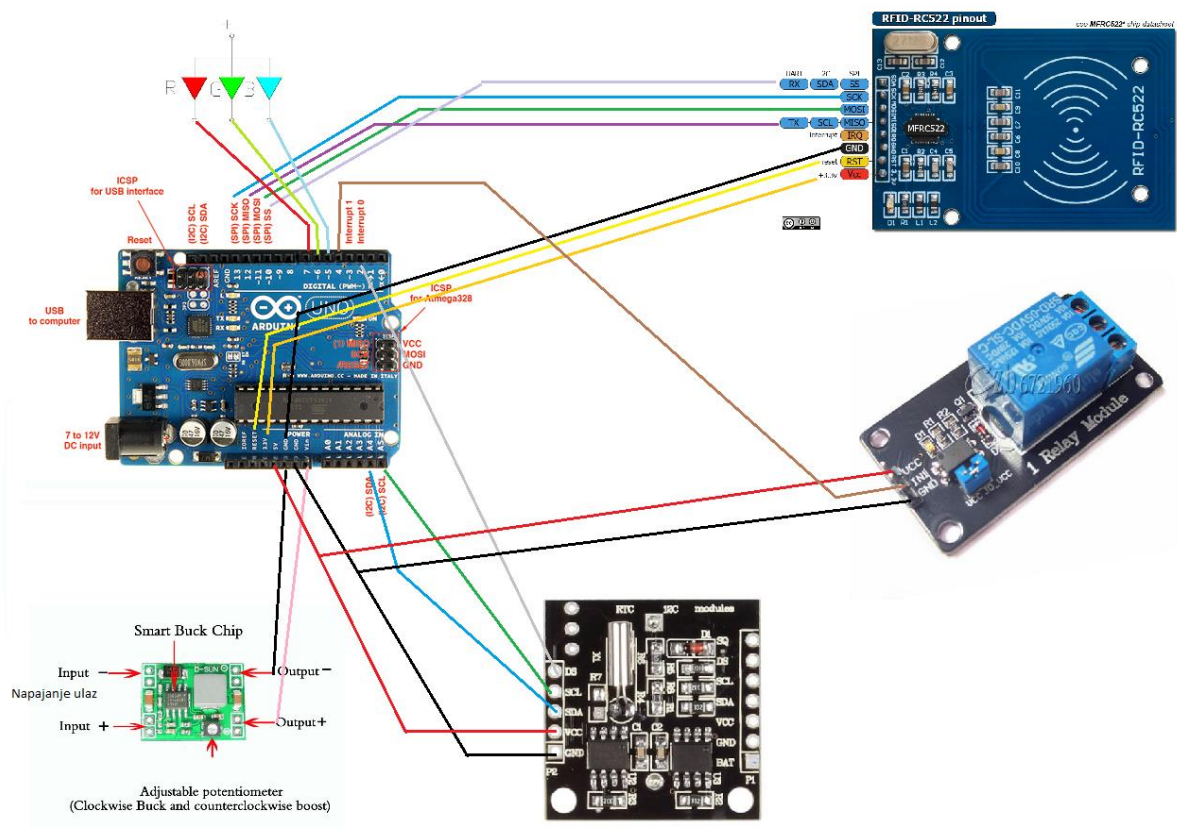
Slika 23. RGB LED dioda

### 3. ELEKTRONIČKA BRAVA

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 UNO pločici i spojiti ih.

#### 3.1. Simulacijski dio

Svakako je preporučljivo napraviti simulaciju gotovog spoja prije početka crtanja tiskane pločice (engl. *Printed Circuit Board (PCB)*). U ovom slučaju korišten je program Proteus ISIS za tu svrhu (kako bi se provjerila ispravnost spoja u električnom smislu). Kao rezultat dobivena je funkcionalna električna shema koja prikazuje međusobni spoj svih modula [Slika 24].

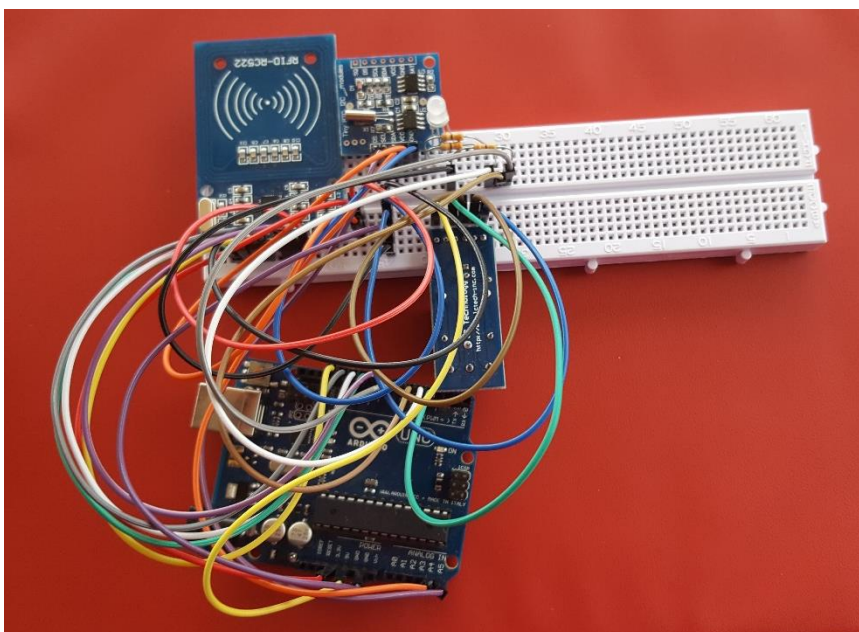


Slika 24. Električna shema spoja svih modula



### 3.2. Eksperimentalni dio

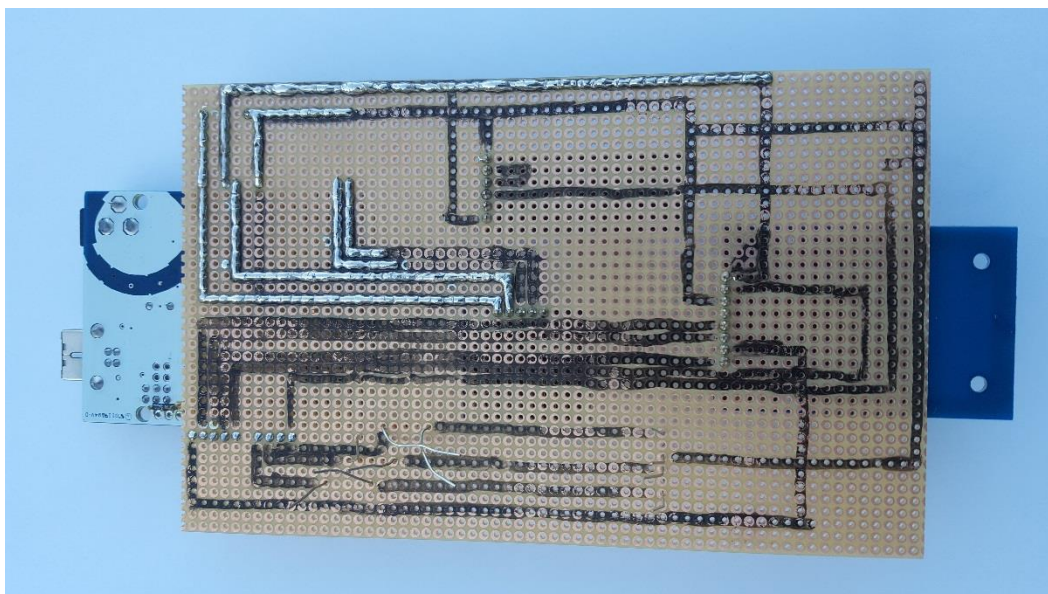
Svaku teoriju i simulaciju potrebno je potvrditi eksperimentom. Svakako prije nego što se može provjeriti ispravnost spojeva i njihova funkcionalnost potrebno je programirati sam mikrokontroler kako bismo odredili protokol komunikacije prema ostalim modulima i definirali sve potrebne funkcije. Taj dio obavljen je u sljedećem poglavlju (poglavlju 4). Nakon programiranja u svrhu potvrde simulacije eksperimentom koristila se eksperimentalna ploča na kojoj su pomoću žica za spajanje, tri otpornika, RGB LED diode i već ranije spomenutih modula prema ranije navedenoj shemi [Slika 24] spojeni svi elementi u cjelinu. Prikazani eksperimentalni spoj može se vidjeti na slici [Slika 25]. Ovime je potvrđen rad samog sklopa nakon dovođenja napajanja na samu Arduino pločicu koji će detaljnije biti opisan u poglavlju programiranja (poglavlje 4).



Slika 25. Električna shema spoja svih modula

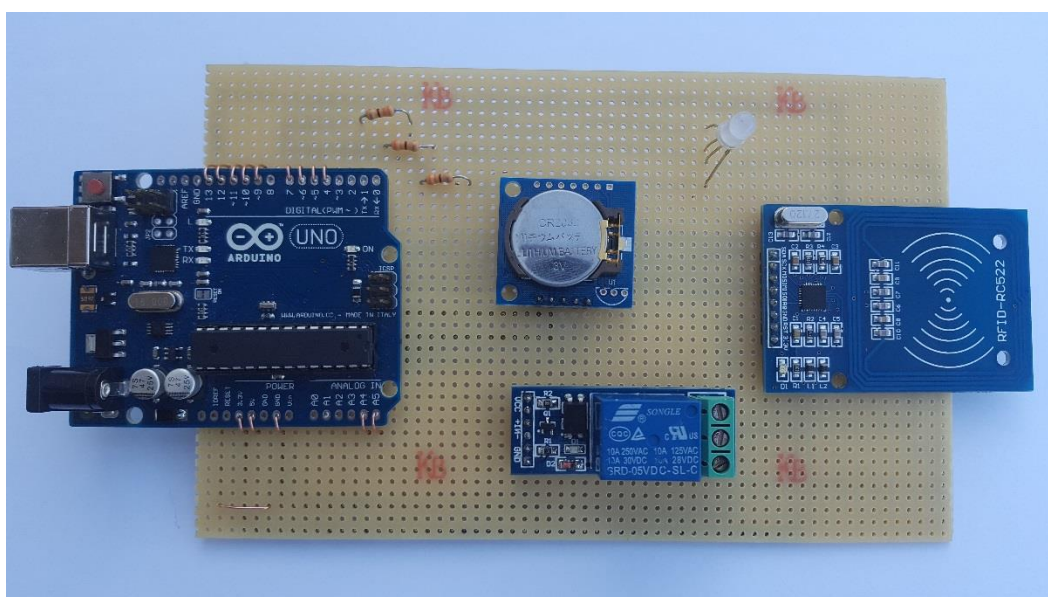
### 3.3. Postupak izrade sklopa

Kako bi se spoj učinio efikasnijim i sigurnijim, a samim time i trajnijim rješenjem, potrebno je ubodne žice zamijeniti zalemljenim spojevima. U tu svrhu korištena je prototipna pločica (engl. *prototype board*) na koju je bilo potrebno postupkom lemljenja spojiti sve potrebne izvođe svih modula i komponenata prema shemi [Slika 24]. Tijek tog postupka spajanja lemljenjem prikazan je sa stražnje strane pločice slikom [Slika 26].



**Slika 26. Prototipna pločica tijekom postupka lemljenja (stražnja strana)**

Prednja strana sa svim potrebnim modulima i žicama za premošćivanje lemnih vodova nakon završenog postupka spajanja prikazana je slikom [Slika 27].

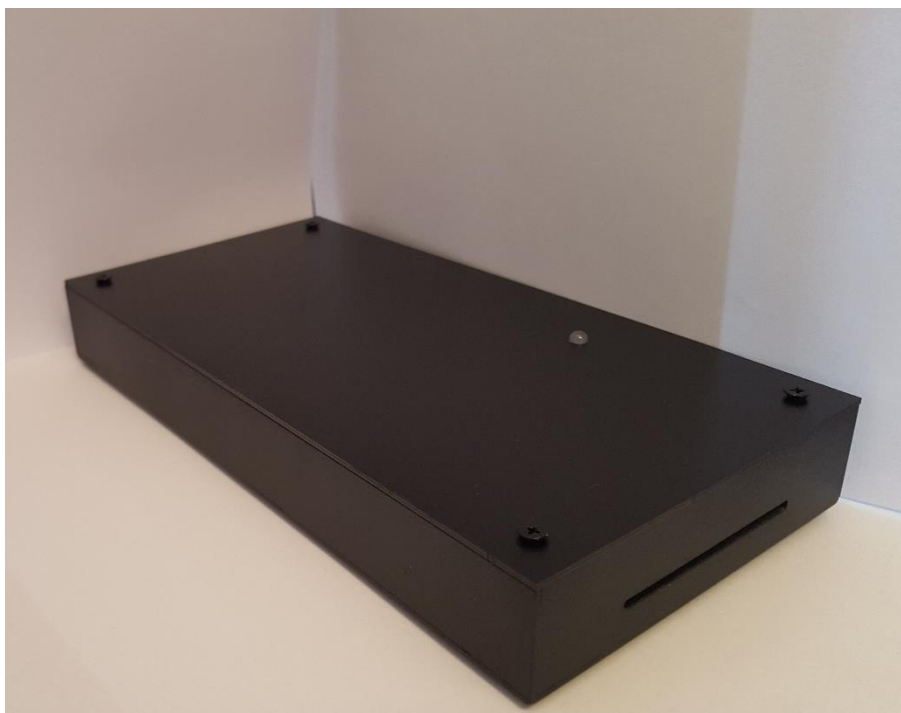


**Slika 27. Prototipna pločica nakon postupka lemljenja (prednja strana)**

Nakon programiranja mikrokontrolera testna faza pokazala se uspješnom na tzv. eksperimentalnoj pločici gdje su pažljivo spojene pojedine komponente i moduli. Nakon provjere rada sve skupa zalemljeno je na posebnu prototipnu pločicu. Kako bi se moglo izraditi kućište u kojem će pojedini moduli i komponente biti spojeni potrebno je izraditi

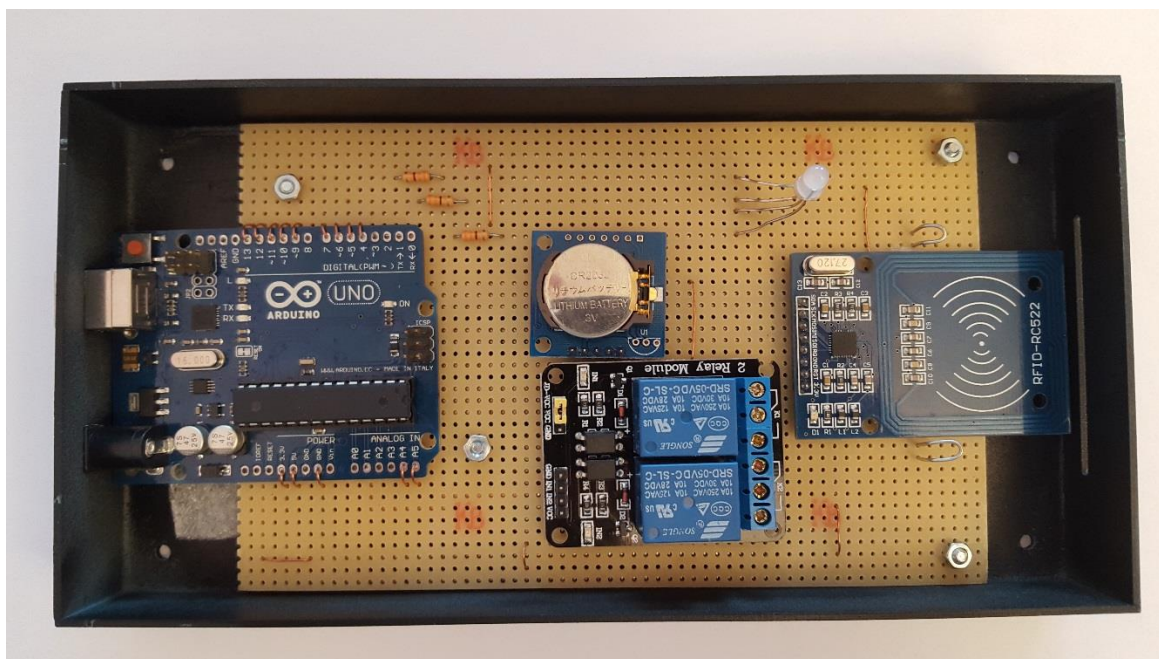
posebnu pločicu koja više nije eksperimentalna nego prototipna pločica (engl. *prototype board*). Nakon pripreme svih komponenata i modula za lemljenje i pripreme same pločice proveden je postupak lemljenja za spajanje na posebnu pločicu bušenog vetronita sa točkama.

Nakon završenog postupka lemljenja potrebno je ispitati učinkovitost i uspješnost samog rada, a to se obavilo pomoću univerzalnog instrumenta pažljivim ispitivanjem svih spojeva (pazeći pritom na kratke spojeve). Zatim su uzete dimenzije samog sklopa kako bi se moglo izraditi kućište. Radi zaštite i kratkog dometa same kartice ukoliko bi bilo problema sa čitanjem kartica s obzirom da je RFID čitač postavljen na dnu kućišta izrađen je u samom kućištu poseban utor za kartice. Također je izbušena rupa za diodu koja signalizira rad uređaja sa gornje strane kućište na samom poklopcu. Gotovo kućište prikazano je slikom [Slika 28].



**Slika 28. Prikaz kućišta s utorom za karticu**

Izvedeno je rastavljivo kućište na način da se poklopac može skinuti. Četiri vijka sa maticom drže poklopac na samom kućištu. Nakon uklanjanja poklopca može se vidjeti pločica koja dimenzijama odgovara samom kućištu [ Slika 29].



**Slika 29. Prikaz kućišta sa otvorenim poklopcem**

Za izradu kućišta korišten je CNC stroj, a materijal od kojeg je kućište izrađeno je vitroplast debljine 1,55 mm. Nakon toga nanescna je temeljna boja u spreju (sive boje) na samo kućište i na kraju autolak crne boje. Dodatne mogućnosti za samo kućište uključivale bi označavanje mjesta gdje se nalazi čitač kartica sa posebnom slikom ili strelicom, kao i dodavanje naziva uređaja na samo kućište pomoću prozirnih naljepnica sa određenom bojom slova. Također je moguće posebno naznačavane USB priključka i priključka za napajanje.

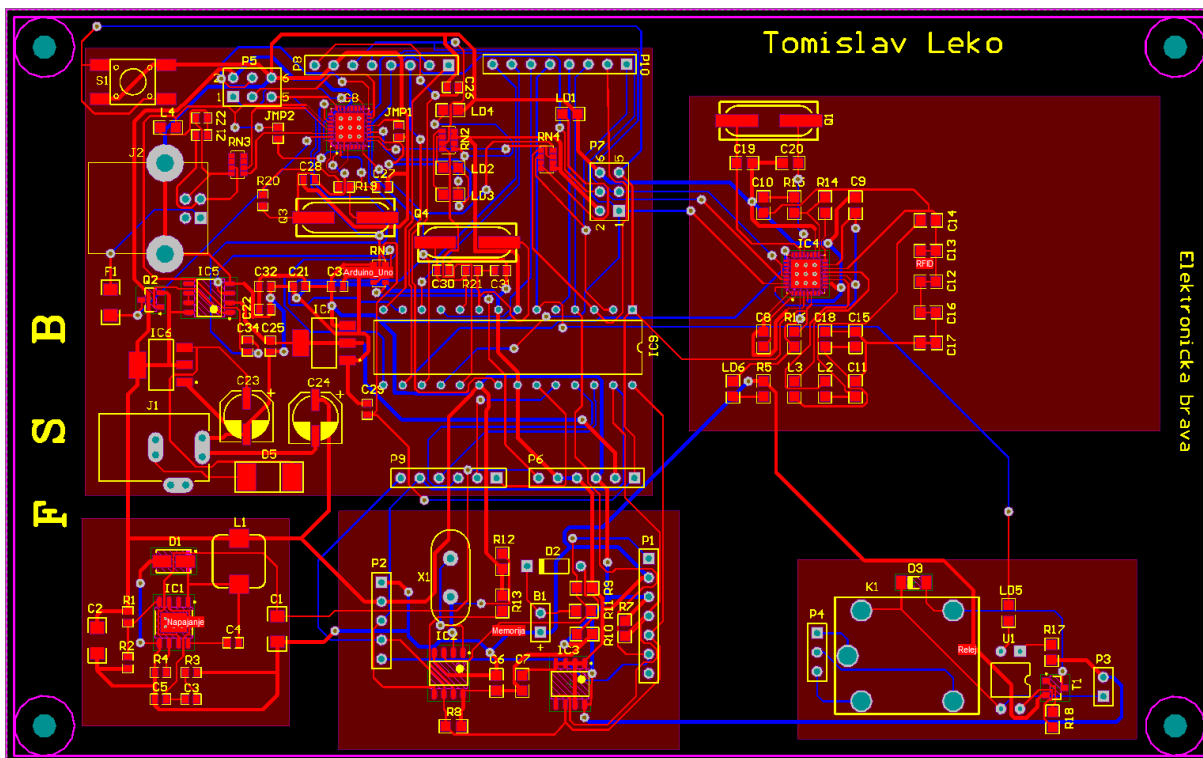
Kompletan sklop sa primjerom jedne kartice programirane za dopušteni ulazak i druge za odbijeni pristup, master karticom u obliku privjeska te kabelom za napajanje prikazani su slikom [Slika 30].



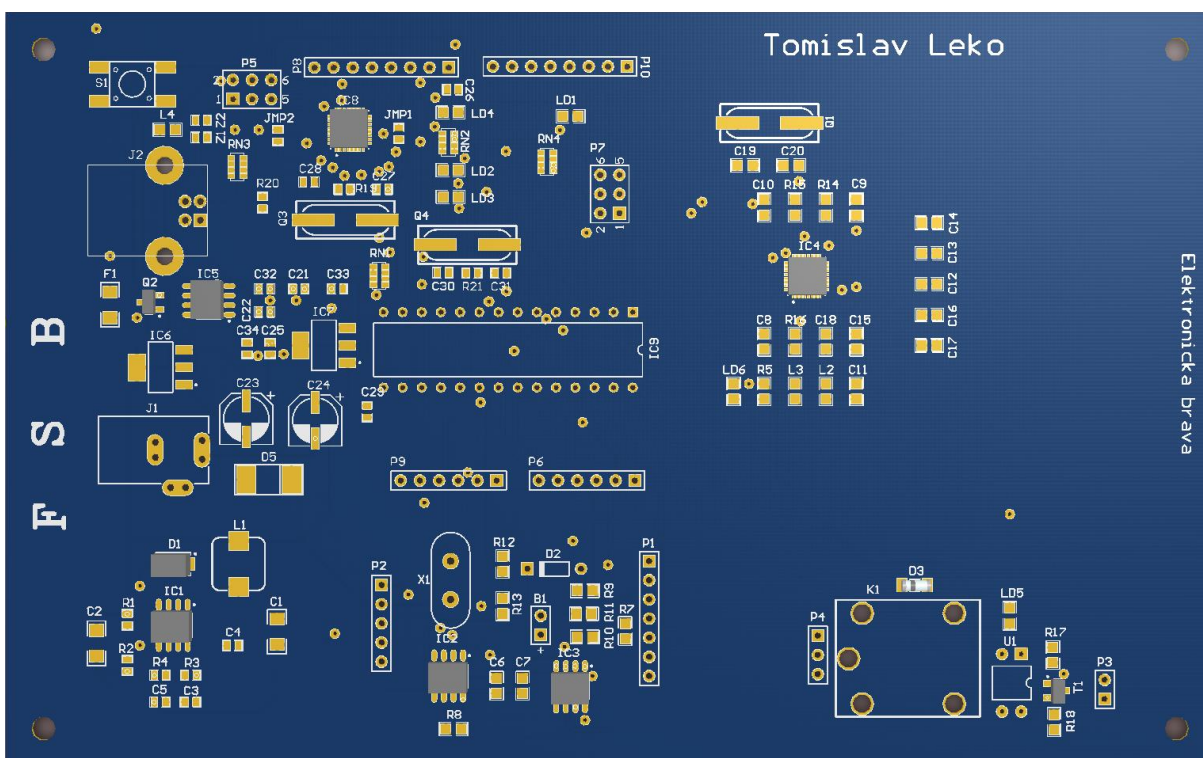
**Slika 30. Prikaz sklopa za demonstraciju rada**

### **3.4. Alternativni postupak izrade**

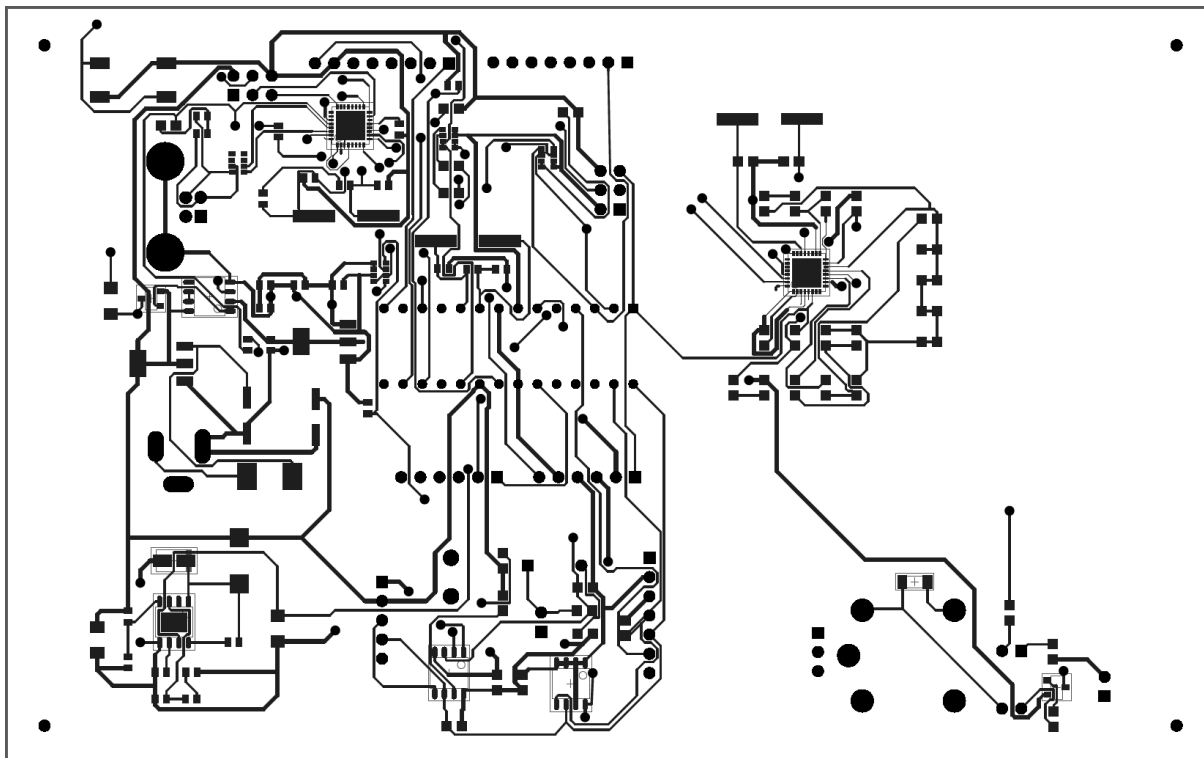
Jedna od alternativa postupku lemljenja za spajanje pojedinih gotovih modula jest izrada zasebne tiskane pločice koja objedinjuje sve komponentne svih modula u jednu cjelinu. Na taj način smanjile bi se znatno dimenzije i težina, te olakšala izrada kućišta za sam sklop. U tu svrhu potrebno bi bilo nacrtati PCB (engl. *Printed Circuit Board*) datoteku u jednom od programa za to. Za crtanje shematskih prikaza pojedinih modula, a naknadno i PCB-a, korišten je programski alat Altium Designer (verzija 14.3.9). Rezultat rada prikazan je na sljedećim slikama na različit način (uobičajen 2D prikaz, 3D prikaz, itd.) [Slika 31], [Slika 32], [Slika 33], [Slika 34].



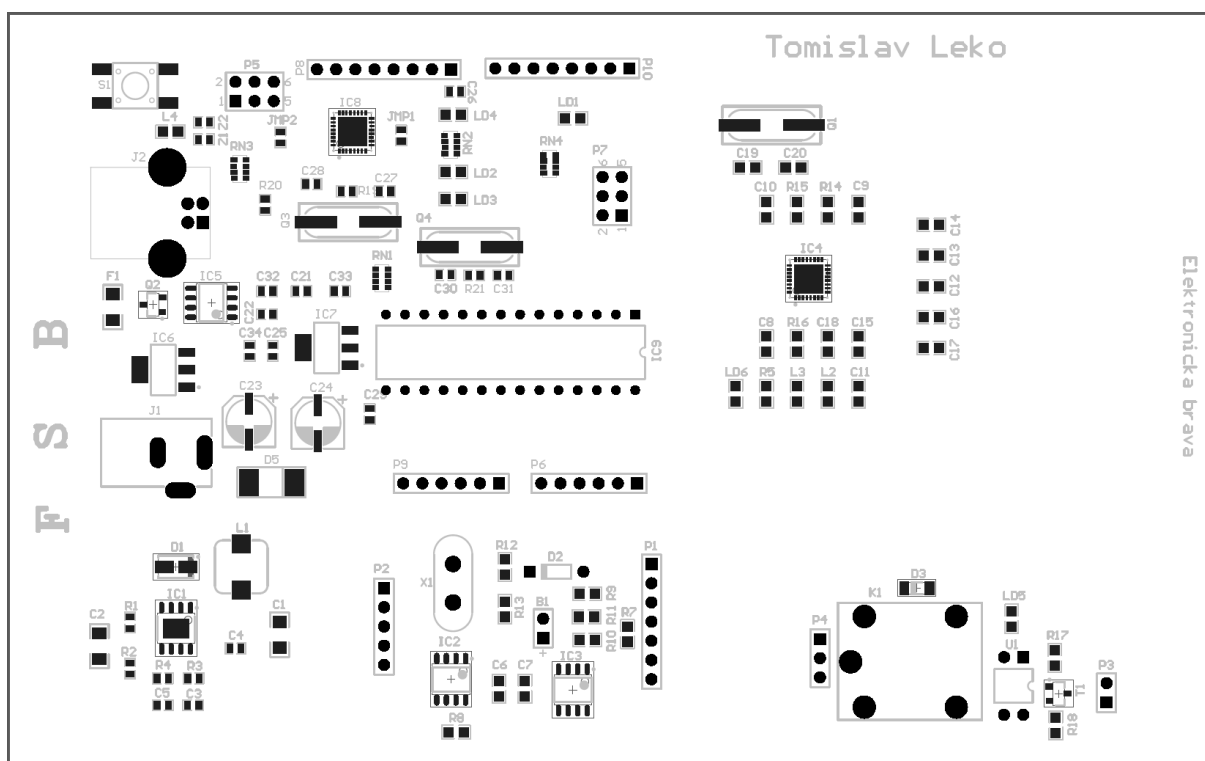
Slika 31. 2D prikaz PCB-a u Altium Designer-u



Slika 32. 3D prikaz PCB-a u Altium Designer-u



Slika 33. Prikaz tiskane pločice (*Final Artwork Print*)



Slika 34. Prikaz prostornog rasporeda komponenta u Altium Designer-u (*Assembly Drawing*)

Sa nacrtanim radom u Altium Designer-u moguće je dati tiskanu pločicu na izradu određenoj firmi koja se bavi projektiranjem, montažom i izradom tiskanih pločica. Ovaj dio izrade

zasebne pločice na kojoj ne bi bili priključeni gotovi moduli već bi se oni već nalazili na toj pločici u obliku komponenata je preskočen radi cijene izrade koja bi bila znatno skuplja nego kupovina već gotovih modula, kao i mogućnosti greške radi kompleksnosti cijelog postupka crtanja. Ali svakako vrijedi napomenuti da postoji mogućnosti izrade jedne pločice što bi svakako bilo estetski i konstrukcijski bolje rješenje (manje dimenzije same kutije sklopa na kraju, kompaktnija izvedba i sl.).

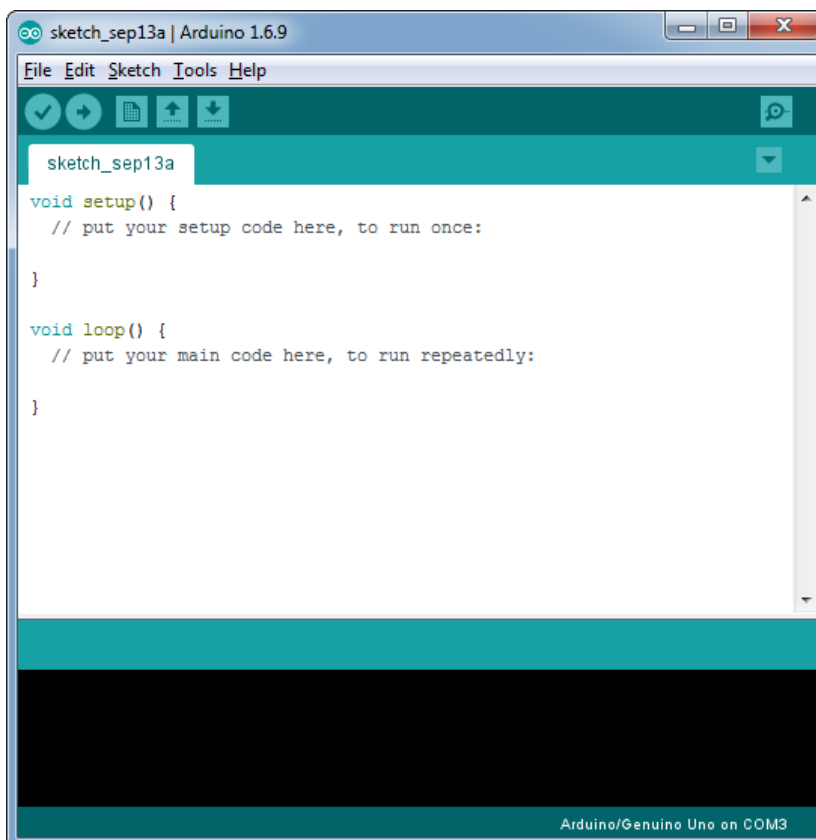


## 4. PROGRAMIRANJE

Nakon spajanja izabranih modula i komponenti te rješavanja električkog dijela sustava da bi se ispitala valjanost rada samog sklopa kao cjeline potrebno je programirati mikrokontroler da odradi zadanu zadaću. Ona uključuje upute putem naredbi samom mikrokontroleru da komunicira sa ostalim modulima i iskoristi njihove mogućnosti u svrhu otvaranja brave putem kotve elektromagneta (simulirano sa radom releja), paljenja i gašenja LED diode u svrhu pokazivanja statusa samog uređaja (prepoznata kartica, odbijena kartica, čekanje na programiranje i/ili prepoznavanje nove kartice i sl.) te evidencije prolazaka na eksterni modul EEPROM memorije. Na taj način moguće je regulirati vrijeme prolaska određene kartice pomoću jedinstvenog UID (engl. *Unique Identification*) koda koji sadrži svaka Mifare RFID kartica, a sastoji se od 8 znamenki.

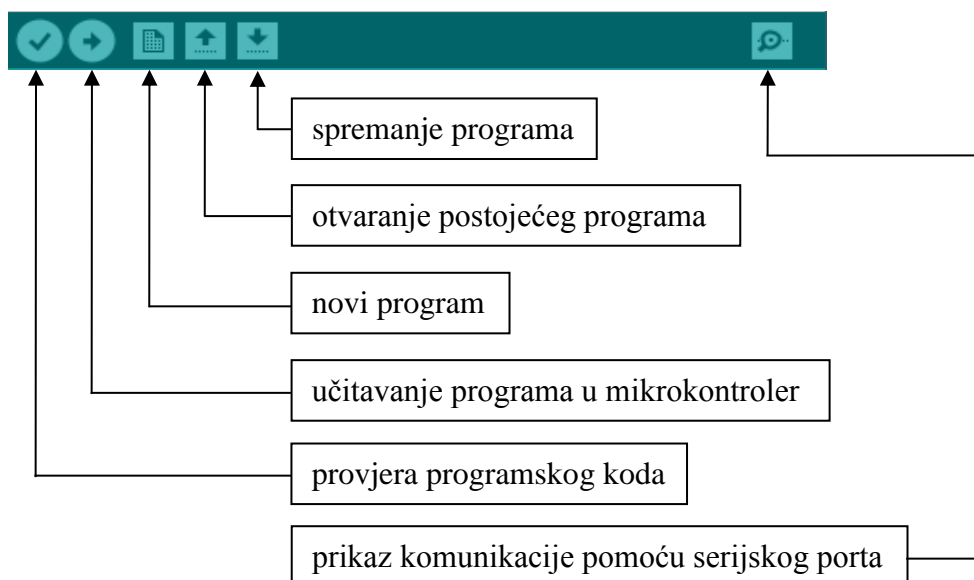
U svrhu programiranja mikrokontrolera korišteno je sučelje dano od strane proizvođača za sam proizvod (Atmega328), a to je Arduino IDE programsko sučelje. Program se piše u modificiranom C++ programskom jeziku prilagođenom upravo Arduino platformi i putem pisanja programskog koda ga prenosimo u mikrokontroler. Program sadrži upute za izvršenje njegove zadaće koristeći ostale komponente samog sustava i upravljanja njihovim radom.

Program za pisanje programa za Arduino mikrokontrolere preuzet je i instaliran sa web-adrese [www.arduino.cc](http://www.arduino.cc). Trenutno aktualna verzija programa u vrijeme pisanja ovog rada je 1.6.9. Prikaz programa nakon pokretanja prikazan je na slici [Slika 35].



**Slika 35. Arduino IDE sučelje**

Prikaz programa možemo podijeliti na nekoliko osnovnih dijelova. Gornji dio programa sastoji se od alatne trake sa izbornikom te dodatnih ikona za brzi dolazak do često korištenih opcija. Srednji dio programa rezerviran je za pisanje koda, dok je u donjem dijelu smještena konzola. Konzola služi za prikaz statusa programa te za povratne poruke o uspješnosti prijenosa napisanog programa u mikrokontroler ili za poruke o detektiranim greškama u napisanom programu. Izbornik s prečacima prikazan je s detaljnim objašnjenjima pojedinih ikona na slici [Slika 36].

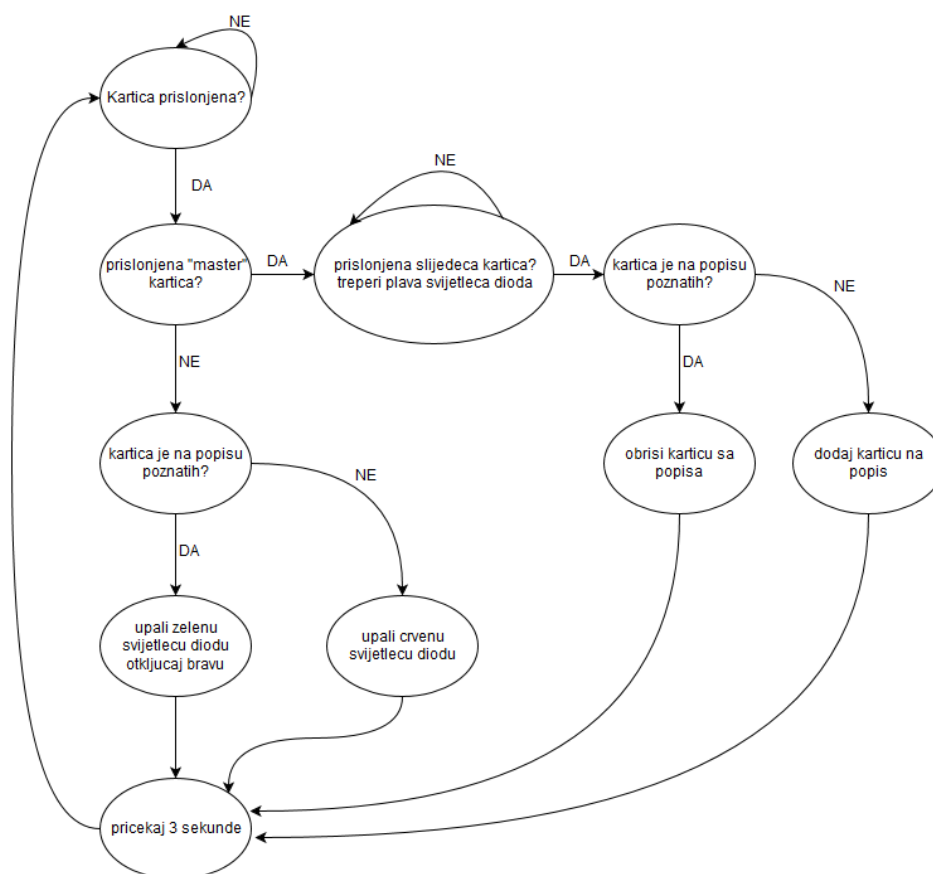


**Slika 36. Izbornik s prečacima**

Nakon preuzimanja i instalacije programa potrebno je podesiti neke od osnovnih parametara u samom programu. Potrebno je odabrati koja Arduino pločica se koristi, na koji priključak računala je ona spojena te način na koji želimo slati programe na sam mikrokontroler. U ovom slučaju odabrana je Arduino UNO pločica za projekt i serijski priključak COM1 na koji je ona spojena. Ova podešenja je potrebno napraviti samo jednom ukoliko se ne mijenja pločica ili korišteni priključak računala. Način na koji možemo saznati koji priključak odabrati jest da odemo u upravitelj uređaja (engl. *Device manager*) i pronađemo Arduino UNO pločicu nakon što smo je povezali sa računalom. Kraj nje stoji oznaka s nazivom serijskog priključka Arduino pločice. Na kraju je potrebno odabrati programator koji koristimo. U ovom slučaju odabrana je varijanta „*Arduino as ISP*“.

#### 4.1. Logika programiranja

Osmišljena je logika kojom će program izvoditi naredbe prikazana na slici [Slika 37].



Slika 37. Logika za programiranje

Program je zamišljen da čeka uputu korisnika držeći pritom signal za čekanje nove kartice (kao indikator svijetli plava LED dioda sve dok se ne prisloni kartica). Ukoliko je prislonjena kartica provjeriti će se da li je prislonjena *master* kartica ili ne. *Master* kartica je obična kartica odabrana i identificirana u programskom kodu da bude *master* kartica pomoću koje će se dodavati nove kartice na popis dozvoljenih ili izbrisati postojeće kartice s popisa. Ako nije prislonjena *master* kartica vrši se provjera da li prislonjena kartica ima pristup ili nema. Ukoliko ima pristup i prepoznata je kao postojeća kartica pali se relej (simbolizira otvaranja vrata) i kao indikator odobrenog prolaza svijetli zelena dioda. To traje tri sekunde i nakon toga se vraća na prvobitno stanje (svijetli plava LED dioda čekajući korisnika da prisloni karticu). Ako kartica nema pristup relej se ne pali (vrata ostanu zaključana) i pali se kao indikator crvena dioda za odbijen pristup. To traje tri sekunde i nakon toga se vraća na prvobitno stanje. Ukoliko se prepoznala na početku prislonjena *master* kartica logika je sljedeća – čeka se iduća kartica od korisnika (kao indikator se pali i gasi plava LED dioda sve do prislanjanja nove kartice) nakon čega se vrši provjera je li dotična kartica na popisu ili nije. Ukoliko nije bila na popisu biti će dodana, a ukoliko je bila, obrisat će se s popisa. Ovo

je jednostavna logika za prikaz kontrole pristupa u neki objekt koja bi se svakako mogla dodatno poboljšati uz znatno kompliciraniju logiku.

## 4.2. Kontrola pristupa

Dodatak uz logiku jest bilježenja vremena ulaska i izlaska iz određenog objekta. Svaka kartica (bez obzira imala pristup ili ne) se bilježi preko svog UID-a skupa sa vremenom i datumom u kojem je prislonjena i ispisuje na zaslone. Na taj način se može vršiti provjera koji korisnik je kada ušao/izašao iz objekta ili pokušao ući ukoliko nema više pristup. To bi se dodatno moglo proširiti na zajedničku bazu podataka i server s kojeg bi se jednostavnim pristupom računala moglo pratiti i slagati izlivanje u sređeni popis.

## 4.3. Programski kod

Zatim je potrebno prevesti logiku rada u programski kod. U prvom dijelu programa se definiraju varijable, odnosno govorimo mikrokontroleru na koji smo izvod što spojili. Koriste se gotove biblioteke za RFID čitač i SPI komunikaciju te definiraju varijable za komunikaciju mikrokontrolera prema modulu i obratno, te za *clock* kojim mikrokontroler upravlja sa samim modulom [Slika 38].

```
RFID_brava
// RFID reader
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN      9           // Configurable, see typical pin layout above
#define SS_PIN       10          // Configurable, see typical pin layout above
MFRC522 mfc522(SS_PIN, RST_PIN); // Create MFRC522 instance
```

Slika 38. Definiranje varijabli – RFID

Digitalni pinovi definirani su zasebnim varijablama radi preglednosti koda i lakšeg praćenja logičnosti istog. Na taj način su crvena, zelena i plava LED svjetla stavljena na digitalne izvođe 7, 6 i 5, a relej (koji predstavlja bravu) je na digitalnom izvodu 4 [Slika 39].

```
// digital pins
#define LED_RED      7
#define LED_GREEN    6
#define LED_BLUE     5
#define LOCK         4
```

Slika 39. Definiranje varijabli – digitalni izvođi

RTC modul koristi I2C komunikaciju koja je nešto drugačija od SPI komunikacije. Tu ne postoje dodatne linije preko kojih se direktno može vidjeti s kime mikrokontroler komunicira već je to riješeno unutar samog protokola. Svaka periferija koja je spojena ima svoju specifičnu adresu preko koje se referencira komunikacija sa mikrokontrolerom. To je u ovom slučaju definirano *Wire* komunikacijom (koja predstavlja I2C komunikaciju). Također je programiran RTC [Slika 40].

```
// rtc
#include <Wire.h>
#define DS3231_I2C_ADDRESS 0x68
```

**Slika 40. Definiranje varijabli – RTC**

Zatim je uključena biblioteka od samog EEPROM dijela sa istog modula. Definirana je *master* kartica preko svog jedinstvenog UID koda [Slika 41].

```
// eeprom
#include <EEPROM.h>
int eeprom_uid_nr = 0;
byte master_uid[4] = {0x29,0xC2,0x17,0xC5}; // hardcoded master card
```

**Slika 41. Definiranje varijabli – EEPROM**

Nakon definiranja varijabli i uključivanja potrebnih biblioteka za pojedini modul definirane su funkcije samih modula. Tijekom programiranja korištene su već gotove funkcije za određene module prilagođene i modificirane za sam kod. Tako je prvo za RTC modul definiran načina računanja vremena i korišten adekvatan binarni zapis za to. Ovim dijelom su uključene i prijestupne godine koje omogućavaju praćenje pravog vremena dugi niz godina. Cijeli dio koda koji se odnosi na taj dio prikazan je slikom [Slika 42].

```

////////////////////////////////// RTC functions ////////////////////////////////////
// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
    return( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
    return( (val/16*10) + (val%16) );
}

void setDS3231time( byte second, byte minute, byte hour, byte dayOfWeek,
                  byte dayOfMonth, byte month, byte year)
{
    // sets time and date data to DS3231
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set next input to start at the seconds register
    Wire.write(decToBcd(second)); // set seconds
    Wire.write(decToBcd(minute)); // set minutes
    Wire.write(decToBcd(hour)); // set hours
    Wire.write(decToBcd(dayOfWeek)); // set day of week (1=Sunday, 7=Saturday)
    Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
    Wire.write(decToBcd(month)); // set month
    Wire.write(decToBcd(year)); // set year (0 to 99)
    Wire.endTransmission();
}

void readDS3231time( byte *second, byte *minute, byte *hour, byte *dayOfWeek,
                   byte *dayOfMonth, byte *month, byte *year)
{
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set DS3231 register pointer to 00h
    Wire.endTransmission();
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
    // request seven bytes of data from DS3231 starting from register 00h
    *second = bcdToDec(Wire.read() & 0x7f);
    *minute = bcdToDec(Wire.read());
    *hour = bcdToDec(Wire.read() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.read());
    *dayOfMonth = bcdToDec(Wire.read());
    *month = bcdToDec(Wire.read());
    *year = bcdToDec(Wire.read());
}

void displayTime ()
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    // retrieve data from DS3231
    readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
&year);
    // send it to the serial monitor
    Serial.print(hour, DEC);

```

```
// convert the byte variable to a decimal number when displayed
Serial.print(":");
if (minute<10)
{
    Serial.print("0");
}
Serial.print(minute, DEC);
Serial.print(":");
if (second<10)
{
    Serial.print("0");
}
Serial.print(second, DEC);
Serial.print(" ");
Serial.print(dayOfMonth, DEC);
Serial.print("/");
Serial.print(month, DEC);
Serial.print("/");
Serial.print(year, DEC);
Serial.print(" Day of week: ");
switch(dayOfWeek) {
case 1:
    Serial.println("Sunday");
    break;
case 2:
    Serial.println("Monday");
    break;
case 3:
    Serial.println("Tuesday");
    break;
case 4:
    Serial.println("Wednesday");
    break;
case 5:
    Serial.println("Thursday");
    break;
case 6:
    Serial.println("Friday");
    break;
case 7:
    Serial.println("Saturday");
    break;
}
}
```

**Slika 42. RTC funkcije**

Zatim su definirane funkcije same EEPROM memorije za brisanje sadržaja, upisivanje novog sadržaja i logika rasporeda UID koda u samu memoriju [Slika 43].



```
////////// eeprom functions //////////////////////////////////////
// delete contents of EEPROM
void delete_eeprom(){
  for(int i = 0; i<EEPROM.length(); i++){
    EEPROM.write(i, 0);
  }
}

// write EEPROM contents to serial port
void list_eeprom_to_serial(void){
  Serial.println(EEPROM.read(0));
  for (int i = 0; i < 5; i++){
    Serial.print(EEPROM.read(4*(i+1)));
    Serial.print(EEPROM.read(4*(i+1)+1));
    Serial.print(EEPROM.read(4*(i+1)+2));
    Serial.println(EEPROM.read(4*(i+1)+3));
  }
}

void write_uid_to_eeprom(byte *uid, byte uid_loc){
  for(int i = 0; i < 4; i++){
    EEPROM.write(uid_loc * 4 + i, uid[i]);
  }
}

int find_empty_space_in_eeprom(void){
  for(int uid_loc = 0; uid_loc < eeprom_uid_nr; uid_loc++){
    for(int byte_loc = 0; byte_loc < 4; byte_loc++){
      if(EEPROM.read((uid_loc + 1)*4 + byte_loc) != 0){
        break;
      }
      return uid_loc + 1;
    }
  }
  return eeprom_uid_nr + 1;
}

void delete_uid_in_eeprom(int uid_loc){
  for(int i = 0; i < 4; i++){
    EEPROM.write((uid_loc * 4 + i), 0); // write zeroes to uid location
  }
  if(uid_loc == eeprom_uid_nr){ // do we have to decrement uid counter
    eeprom_uid_nr--;
    EEPROM.write(0, eeprom_uid_nr);
  }
}
```

```
int uid_present_in_eeprom(byte *uid){
    int uid_present_flag = 0;

    for(int uid_nr = 0; uid_nr < eeprom_uid_nr; uid_nr++){
        uid_present_flag = 1;
        for(int i = 0; i < 4; i++){
            byte tmp_uid_byte = EEPROM.read((uid_nr + 1) * 4 + i);
            if( mfrc522.uid.uidByte[i] != tmp_uid_byte){
                uid_present_flag = 0;
                break;
            }
        }
        if(uid_present_flag) { // card uid found in eeprom
            return uid_nr + 1 ; //
        }
    }
    return 0; // uid not found
}
```

**Slika 43. EEPROM funkcije**

Zatim započinje glavni dio programa. Svaki program u Arduino okolišu se sastoji od dvije osnovne funkcije; jedna je *setup* koja se izvrši jednom na početku samog izvršavanja programa, a druga je *loop* koja kada dođe do kraja ponovno ju pozove okoliš u mikrokontroler i na taj način se vrti u tzv. petlji. U *setup* dijelu programa se izvršila inicijalizacija svih dodatnih mogućnosti koje su spojene na mikrokontroler i koriste se (kao npr. *Wire.begin* koja je inicijalizacijska funkcija za I2C priključak). Na taj način je postavljen i inicijaliziran RTC modul, digitalni izlazni izvodi, serijski port, RFID čitač i pročitan je podatak iz EEPROM memorije o broju trenutno zapisanih kartica. Ovaj dio koda prikazan je slikom [Slika 44].

```
void setup() {
// RTC INIT
  Wire.begin();
  // set the initial time here:
  // DS3231 seconds, minutes, hours, day, date, month, year
  //setDS3231time(10,24,16,4,3,8,16);

// DIGITAL PINS INIT
  pinMode(LED_RED, OUTPUT);
  pinMode(LED_GREEN, OUTPUT);
  pinMode(LED_BLUE, OUTPUT);
  pinMode(LOCK, OUTPUT);

// UART INIT
  Serial.begin(9600); // Initialize serial communications with the PC
  while (!Serial); // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
// RFID INIT
  SPI.begin(); // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522
  //Serial.println(F("MFRC522 firmware version"));
  //mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD - MFRC522 Card Reader details

// EEPROM INIT
  eeprom_uid_nr = EEPROM.read(0);
  //delete_eeprom();
  //list_eeprom_to_serial();
}
```

**Slika 44. Glavni dio programa (*setup*)**

U drugom dijelu (*loop*) pomoću funkcije RFID čitača uspoređujemo da li je nova kartica prislonjena ili nije i tako preko naredbe *digitalWrite* postavljamo određene izvođe za LED diodu ili relej u visoko (engl. *HIGH*) ili nisko (engl. *LOW*) stanje. I ovaj dio programa se stalno vrti i provjerava prisutnost nove kartice i daje adekvatan prikaz pomoću LED diode o trenutnom stanju i po potrebi okida relej. Dio koda koji se odnosi na ovo prikazan je slikom [Slika 45].

```

// main program logic
void loop() {
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_BLUE, HIGH);
    digitalWrite(LOCK, LOW);
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial() ) {
    return;
  }
  mfrc522.PICC_HaltA(); // wait until the card is removed from reader to read again

  // check if card is master card
  int master_flag = 1;
  for(int i = 0; i < 4; i++){
    if(mfrc522.uid.uidByte[i] != master_uid[i]){
      master_flag = 0;
      break;
    }
  }

  if( master_flag) { // master card present
    while(!mfrc522.PICC_IsNewCardPresent()){
      digitalWrite(LED_BLUE, !digitalRead(LED_BLUE));
      digitalWrite(LED_RED, LOW);
      digitalWrite(LED_GREEN, LOW);
      delay(100);
    }; // wait for new card
    digitalWrite(LED_BLUE, HIGH);
    mfrc522.PICC_ReadCardSerial();
    mfrc522.PICC_HaltA(); // wait until the card is removed from reader to read again

    int uid_known = uid_present_in_eeprom(mfrc522.uid.uidByte); // is UID in known UID list?
    if( uid_known ){ // card UID known -> delete it
      delete_uid_in_eeprom( uid_known );
    } else { // card UID not known -> add it to the list
      int empty_loc = find_empty_space_in_eeprom();
      //Serial.print(empty_loc);
      //list_eeprom_to_serial();
      if( empty_loc > eeprom_uid_nr){
        eeprom_uid_nr++;
        EEPROM.write(0, eeprom_uid_nr);
      }
      //list_eeprom_to_serial();
      write_uid_to_eeprom(mfrc522.uid.uidByte, empty_loc);
      //list_eeprom_to_serial();
    }
  }
}

```

```
    } else { // regular card present
        // write card info to serial port
        displayTime();
        Serial.print("UID: ");
        Serial.print(mfrc522.uid.uidByte[0], HEX);
        Serial.print(mfrc522.uid.uidByte[1], HEX);
        Serial.print(mfrc522.uid.uidByte[2], HEX);
        Serial.println(mfrc522.uid.uidByte[3], HEX);
        //list_eeprom_to_serial();
        mfrc522.PICC_HaltA(); // wait until the card is removed from reader to read again

// check if card is present in list of uids
int uid_present_flag = uid_present_in_eeprom( mfrc522.uid.uidByte );
if(uid_present_flag){ // card found in UID list -> open the door
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(LED_BLUE, LOW);
    digitalWrite(LOCK, HIGH);
} else { // card not found in UID list -> do not open the door
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_BLUE, LOW);
    digitalWrite(LOCK, LOW);
}
delay(3000);
}
}
```

**Slika 45. Glavni dio programa (*loop*)**

Logika i funkcioniranje samog sklopa opisana je u uvodnom dijelu programiranja, a ona je realizirana strojnim jezikom uz pomoć već postojećih funkcija za pojedine module i samog Arduino sučelja. Na taj način sklop može obaviti svoju funkciju preko mikrokontrolera koji je pomoću programskog koda upoznat sa svojom svrhom.

## 5. ZAKLJUČAK

U današnje vrijeme razvoja elektronike i tehnika izrade tiskanih pločica znatno su lakše mogućnosti ostvarivanja vlastitih ideja. Proučavajući sklopove i programe koje su drugi osmislili, te kroz dostupan materijal na Internetu i iskustva raznih hobista, kao i djela ozbiljnijih projekata, moguće je dobiti dobru osnovu za ulazak u zanimljiv i intrigantan svijet mikrokontrolera koji nam pružaju neograničene mogućnosti. Stoga možemo reći da ovaj način uporabe navedenih sustava može predstavljati značajan napredak kako u poslovnom svijetu tako i u svakodnevnom životu.

Beskontaktne brave mogu se činiti kao nepotreban luksuz ali primjena na područjima automatskog upravljanja i hotelskih sustava, sustava podizanja rampi za ulaske/izlaske u određene objekte, olakšani česti ulazak/izlazak iz prostorije bez potrebe nošenja mehaničkog ključa sa sobom može u velikom zanosu predstaviti novo i modernije doba. Naposljetku tehnika je i kroz povijest napredovala upravo radi potrebe da si olakšamo život i stvorimo si lakše i ugodnije okruženje.

Najveći nedostatak opisanog sklopa za beskontaktno otključavanje brave 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 zloupotrebjavati tako da nije preporučljivo primijeniti ga u nekim institucijama gdje se čuvaju važni i osjetljivi podaci bez nekakvih dodatnih zaštitnih mjera i sustava.

Prednosti su već nekoliko puta spomenute kroz ovaj rad, ali glavna je jednostavno i brže ulaženje i izlaženje iz prostorije bez potrebe za nošenjem mehaničkog ključa sa sobom. Na taj način ulazimo u moderno i sofisticirano doba koje ionako sa sobom nosi sve jednostavnije stvari u tehnici sa što jednostavnijim pristupom za korisnike dok je onaj inženjerski dio stavljen upravo unutar tzv. „crne kutije“.

## LITERATURA

- [1] Zenzerović, P.: Arduino kroz jednostavne primjere, Hrvatska zajednica tehničke kulture, Zagreb, 2014.
- [2] Lukić, B.; Milutinović, I.: Tehnologija spajanja materijala lemljenjem, AGM knjiga, 2015.
- [3] Platt, C.: Make: Electronics 2nd Ed., Maker Media, 2015.
- [4] Herold, Z.: Računalna i inženjerska grafika, Zagreb, 2003.
- [5] <http://www.doityourself.com/>
- [6] <https://www.arduino.cc/>
- [7] <https://www.sparkfun.com/>

## **PRILOZI**

- I. CD-R disc
- II. Tehnička dokumentacija