

# Registracija studenata na nastavi pomoću X-ica

---

**Putanec, Patrik**

**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:235:717567>

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

*Download date / Datum preuzimanja:* **2024-07-24**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Patrik Putanec**

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentori:

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

Student:

Patrik Putanec

Zagreb, 2022.

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

*Zahvaljujem se svom mentoru, prof. dr. sc. Mladenu Crnekoviću koji mi je svojim znanjem i iskustvom pomagao u realizaciji praktičnog dijela završnoga rada.*

*Zahvaljujem se kolegi Filipu Čaviću na elektroničkim komponentama koje mi je posudio.*

*Posebno se zahvaljujem svojim roditeljima na razumijevanju i velikoj podršci tijekom cijelog preddiplomskoga studija.*

**Patrik Putanec**



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne i diplomske 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: **Patrik Putanec**

JMBAG: **0035219628**

Naslov rada na hrvatskom jeziku: **Registracija studenata na nastavi pomoću X-ica**

Naslov rada na engleskom jeziku: **Students registration in class using X cards**

Opis zadatka:

Potrebno je projektirati uređaj za osobnu registraciju studenata na nastavi. Prijava studenata vršila bi se preko X-ice, a podaci bi se bilježili na odgovarajućem mediju. Nakon nastave podaci bi se prebacivali na osobno računalo. Sučelje na mobitelu trebalo bi pokazati popis trenutno registriranih studenata, a sučelje na osobnom računalu trebalo bi pokazati ukupnu statistiku dolazaka za svakog studenta. Povezivanje uređaja s mobitelom i osobnim računalom se ostvaruje Bluetooth komunikacijom.

U radu je potrebno:

- projektirati uređaj za osobnu registraciju preko X-ice,
- odabrati potrebne senzore, signalizaciju i upravljački mikrokontroler,
- projektirati upravljački PCB i predložiti strukturu programa,
- predložiti sučelje na osobnom računalu i mobitelu,
- procijeniti vrijednost uređaja.

Potrebno je navesti korištenu literaturu, te eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2021.

Datum predaje rada:

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

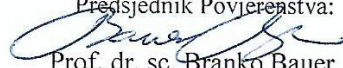
Predviđeni datumi obrane:

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

Zadatak zadao:

  
Prof. dr. sc. Mladen Crneković

Predsjednik Povjerenstva:

  
Prof. dr. sc. Branko Bauer

## SADRŽAJ

SADRŽAJ .....	1
POPIS SLIKA .....	2
POPIS TABLICA.....	3
SAŽETAK.....	4
SUMMARY .....	5
1. UVOD.....	6
1.1. Ideja sustava .....	6
2. ČITAČ KARTICE .....	7
2.1. Magnetska traka i magnetski čitač .....	7
2.2. NFC čip i RFID čitač .....	8
2.3. Odabir čitača .....	9
2.4. PN532 NFC RFID modul .....	9
2.4.1. I2C komunikacija.....	10
2.5. Vanjska memorija .....	11
2.5.1. SPI komunikacija .....	11
2.6. RTC sklop .....	12
2.7. Mikroupravljač.....	13
2.8. Bluetooth komunikacija .....	15
2.8.1. UART komunikacija .....	15
2.9. Napajanje .....	16
2.10. LED u boji.....	16
2.11. Vizualna potvrda .....	17
3. Shema pločice uređaja .....	18
4. Idejni program uređaja.....	20
5. Program uređaja.....	24
5.1. Raspored predavanja .....	24
5.2. Baza studenata.....	25
5.3. Primjer datoteke predavanja.....	26
5.4. Verifikacija programa .....	28
6. Procjena vrijednosti uređaja .....	32
7. ZAKLJUČAK.....	33
LITERATURA.....	34
PRILOZI.....	35
PRILOG II.....	36

## POPIS SLIKA

Slika 1.	Vizualizacija magnetski pohranjenih podataka, preuzeto sa [2] .....	7
Slika 2.	Konvencionalni čitač magnetske kartice .....	8
Slika 3.	Pasivna beskontaktna kartica, preuzeto sa [3] .....	8
Slika 4.	Elechouse PN532 NFC RFID modul .....	9
Slika 5.	Prijenos informacija pomoću I2C komunikacije, preuzeto sa [9] .....	10
Slika 6.	Modul za mikro SD karticu .....	11
Slika 7.	RTC modul PCF85063A .....	12
Slika 8.	Arduino UNO .....	13
Slika 9.	Pregled pinova UNO pločice, preuzeto sa [13] .....	14
Slika 10.	HC-05 Bluetooth modul .....	15
Slika 11.	Ćelije i 3.7-5V DC-DC konverter .....	16
Slika 12.	RGB dioda, zajednička katoda .....	16
Slika 13.	Električna shema uređaja .....	18
Slika 14.	PCB pločica .....	19
Slika 15.	Render PCB pločice sa konektorima i otpornicima .....	19
Slika 16.	Glavna petlja u programu .....	20
Slika 17.	Naredba "NaDan" .....	21
Slika 18.	Naredba "Tablica" .....	22
Slika 19.	'RULE.txt' -datoteka sa rasporedom predavanja .....	24
Slika 20.	'stud.txt' -baza studenata .....	26
Slika 21.	Primjer datoteke predavanja - 'VJ6.txt' .....	26
Slika 22.	pomoćna 'num.txt' datoteka .....	27
Slika 23.	Naredba 'NaDan' preko mobitela .....	28
Slika 24.	Naredba 'tabl' preko laptopa .....	29
Slika 25.	Ubacivanje podataka u tablicu .....	30
Slika 26.	Razdvajanje podataka na stupce .....	31
Slika 27.	Tablica evidencije za jedan predmet. ....	31

## **POPIS TABLICA**

Tablica 1. Ukupni trošak za komponente ..... 32



## **SAŽETAK**

Nakon objašnjenja ulaznih motivacija za ovu temu, ukratko je opisan uređaj koji bi mogao voditi evidenciju studenata pomoću X-ica. Dano je obrazloženje odabira čitača i prezentirane su sve potrebne komponente za izradu funkcionalnog uređaja. Zadaci uređaja su podijeljeni na manje cjeline te je opisan teorijski tijek programa. Projektiranjem pločice i pisanjem Arduino kôda izrađen je potpuno funkcionalni prototip. Konačno, dani su naputci za daljnji razvoj i unaprjeđenje sustava u slučaju da bude primijenjen u stvarnim uvjetima.

Ključne riječi: beskontaktna kartica, bežični prijenos na male udaljenosti, SD kartica

## **SUMMARY**

After elaborating the motivation for this work, a device that could register students by means of their X-card is briefly described. The choice for a particular type of card reader is elaborated on and all needed components are presented. The device's tasks are decomposed into smaller pieces and a theoretical program flow chart is presented. A fully functional prototype was built after designing the PCB and writing Arduino code. Finally, a few considerations are given for the further enhancement and upgrades should this system be used in real-world conditions.

Key words: smart cards, Near Field Communication, SD card

## 1. UVOD

Provjera prisustva studenata na nastavi još uvijek predstavlja svakodnevnicu u akademskim ustanovama. Unatoč velikoj digitalizaciji obrazovanja i uvođenja "pametnih" rješenja, provjera prisustva se gotovo isključivo provodi ručnim potpisivanjem studenata na papir dan od strane trenutnog nastavnika. Digitalizacija ovog procesa, osim prednosti prebacivanja posla sa nastavnika i studenata na uređaj, potencijalno može i ubrzati postupak. Ovaj rad predstavlja koncept uređaja koji bi funkcionirao sa već postojećom "infrastruktururom" tj. izdanim studentskim iskaznicama.

### 1.1. Ideja sustava

Ideja je u tome da se potpisivanje studenta može obaviti jednostavnim očitanjem njegove studentske X-ice prilikom dolaska na predavanje. Čitanjem kartice dobiva se matični broj studenta koji se upisuje na trenutno predavanje, a sustav bi trebao biti fleksibilan da podržava različita predavanja i samostalno vodi brigu o rasporedu sati. Za obavljanje ovakvog zadatka potrebne su slijedeće komponente:

- Čitač kartice za ostvarivanje interakcije.
- Mikroprocesor koji upravlja uređajem.
- Vanjska memorija za pohranu dnevnika prisustva.
- Sklop za praćenje vremena (RTC - Real Time Clock).
- Sredstvo za vizualnu potvrdu upisa (LCD display ili LED u boji).
- Bluetooth komunikator za prijenos podataka.
- Izvor napajanja za sve podsklopove.

Dnevnik prisustva sastoji se od dokumenta za svako predavanje, a u svakom dokumentu su zasebno zapisani datumi održavanja te studenti koji su bili prisutni. Povezivanjem uređaja sa mobitelom ili laptopom pomoću Bluetooth komunikacije omogućuje se interaktivno sučelje kroz terminal. Upisivanjem dogovorenih naredbi može se zatražiti informacija o studentima prisutnima na nekom predavanju nekog datuma ili se može zatražiti ispis podataka za popunjavanje excel tablice prisutnosti za odabrani predmet/kolegij.

U nastavku slijedi odabir i kratki opis svake od potrebnih komponenti, počevši od najbitnije-čitača kartice.

## 2. ČITAČ KARTICE

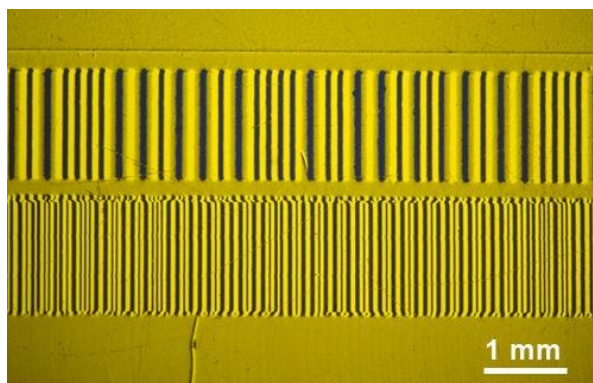
Kako X-ica posjeduje i magnetsku traku i beskontaktni čip, njeno čitanje se može izvršiti na ta dva različita načina:

- čitanjem magnetske trake provlačenjem kartice kroz magnetski čitač, ili
- čitanjem NFC čipa približavanjem kartice prema beskontaktnom RFID čitaču.

U nastavku je dan kratki opis ova dva sustava.

### 2.1. Magnetska traka i magnetski čitač

Magnetska traka predstavlja dio kartice koji sadrži podatke zapisane u obliku naizmjenično polariziranih feromagnetskih čestica. Izrađene su tako da se čestice magnetiziraju u željeno stanje pomoću preciznog elektromagneta dok plastično vezivo ne očvrсне i trajno zadrži čestice u takvom stanju [1].



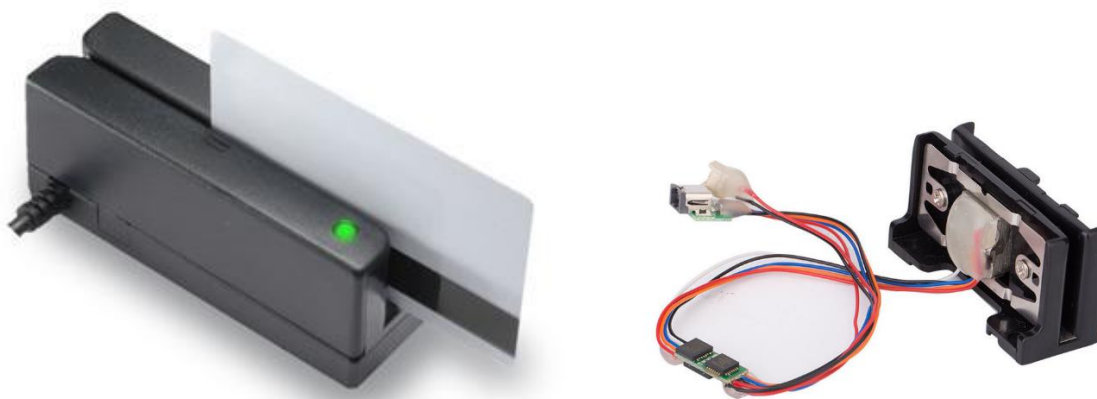
Slika 1. Vizualizacija magnetski pohranjenih podataka, preuzeto sa [2]

Najčešće su izrađene tako da sadrže dvije ili tri paralelne trake od kojih prva ("Track 1") ima najveću gustoću informacija u iznosu od 210 bpi (bits per inch) te sadrži slova i brojeve [2]. Na prvu traku se na standardizirani način upisuju podaci korisnika kartice. U slučaju X-ice to su:

- Dogovoreni header podataka, simboli %B sukladno [2]
- Skup od 8 brojeva koji predstavlja podatke o izdavaču – Sveučilištu
- Skup od 11 brojeva koji predstavlja studentov JMBAG
- Skup slova varijabilne dužine- Prezime i Ime studenta
- OIB studenta te dogovoreni kraj trake, simbol ?

Druga i treća traka (Track 2 i Track 3) imaju gustoću informacija u iznosu od 75 bpi te sadrže samo brojeve [2]. Druga traka na X-ici redundantno sadrži podatke o Sveučilištu i studentov JMBAG dok je treća traka neiskorištena.

Čitanje ovih podataka sa magnetske trake se provodi prolaskom kartice preko osjetljivog detektora magnetskog polja (čitajuće glave) koji prepoznaje spomenute promjene polariteta.



Slika 2. Konvencionalni čitač magnetske kartice

## 2.2. NFC čip i RFID čitač

Beskontaktno čitanje kartice odvija se uz pomoć predajničke antene (čitač) i antene ugrađene u karticu koje tvore sustav sličan transformatoru sa zračnom jezgrom [5]. Čitač odašilje radiovalove koje antena kartice modulira mijenjajući vlastitu otpornost, te se ta promjena može detektirati u anteni čitača [4]. Za modulaciju i prijenos pohranjenih podataka je zadužen čip u kartici. Aktivne kartice imaju vlastitu bateriju a kod pasivnih kartica se potrebna energija dobavlja iz valova čitača. X-ica je pasivna.



Slika 3. Pasivna beskontaktna kartica, preuzeto sa [3]

Količina i vrsta pohranjenih podataka ovisi o vrsti ugrađenog čipa [6]. Kartice mogu sadržavati razne podatke ovisno o primjeni (osobne podatke, podatke za aplikacije, programibilne i šifrirane podatke itd.). X-ice sadrže samo jedinstveni broj kartice, tzv. UID (Unique Identifier) koji nema veze sa podacima korisnika kartice.

### 2.3. Odabir čitača

Velika prednost magnetske trake je činjenica da se njenim čitanjem može odmah odrediti matični broj studenta koji se registrirao, što znači da nije potrebno imati poznatu bazu podataka sa matičnim brojevima studenata koje očekujemo. Prednost beskontaktnog čitača je u jednostavnosti korištenja, no kod njega je potrebno broj kartice (UID) povezati sa identitetom studenta kako bi se izvršio upis (preko baze podataka). Sa stajališta zaštite osobnih podataka magnetski čitač ima prednost, no sa stajališta stvarne implementacije u sustav, beskontaktni čitač je u ovom trenutku bolji. Razlog tome je teško povezivanje klasičnog magnetskog čitača sa mikroupravljačem i nerazmjerno visoka cijena čitača koji ima ugrađeni kompatibilni komunikacijski sustav. Dakle, unatoč nedostatku beskontaktnog čipa X-ice, u radu će biti korišten beskontaktni čitač.

### 2.4. PN532 NFC RFID modul

Odabrani čitač je Elechouse PN532 sa stranice [7].



Slika 4. Elechouse PN532 NFC RFID modul

Neke od tehničkih karakteristika ovog modula izdvojene iz dokumentacije [8]:

- Mogućnost povezivanja sa mikroupravljačem pomoću I2C, SPI ili HSU (High Speed UART) komunikacije.
- RFID čitanje i pisanje standardnih kartica.
- Čitanje kartice na udaljenosti od 5~7cm.
- Napon napajanja 3.3~5V (ovisno o vrsti komunikacije)

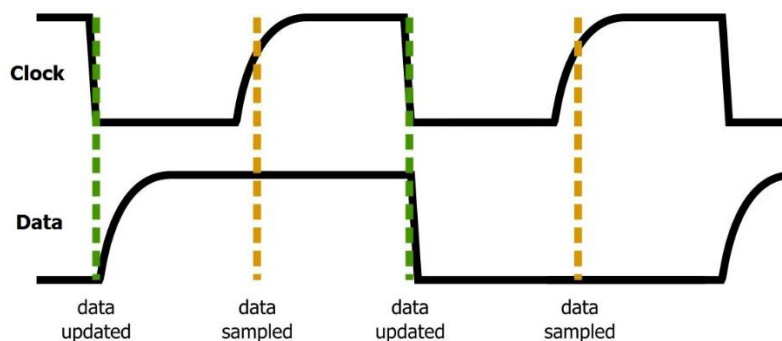
Modul će biti povezan sa mikroupravljačem pomoću I2C komunikacije.

#### 2.4.1. I2C komunikacija

I2C ili IIC (engl. Inter-Integrated Circuit) je vrsta sinkrone serijske komunikacije preko mreže sa jednim od više spojenih uređaja. Komunikacija se odvija preko dva pina:

- SDA (Serial DAta) je zajednički vod preko kojeg se prenose podaci na način da se vod postavlja u logičku nulu ili jedinicu
- SCL (Serial CLock) je zajednički vod preko kojeg se vrši sinkronizacija uzorkovanja podataka sa SDA voda

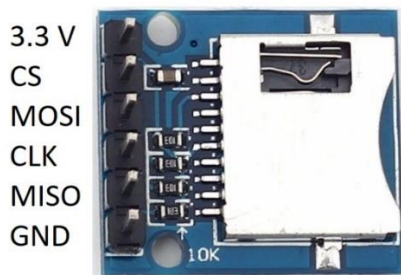
Transmisija podataka započinje slanjem 7-bitne adrese, jedinstvene uređaju kojem se poruka želi poslati (npr. adresa PN532 je 1001000 tj. 0x48). Na taj način jedino uređaj koji je prozvan "sluša" a ostali uređaji na mreži miruju i ne smetaju. Prijenos informacija se događa tako da na svaki padajući brid signala na SCL vodu (kojeg kontrolira glavni uređaj), podređeni uređaj postavlja novi bit informacije na SDA vod. Glavni uređaj to stanje čita na slijedećem rastućem bridu na SCL vodu.



Slika 5. Prijenos informacija pomoću I2C komunikacije, preuzeto sa [9]

## 2.5. Vanjska memorija

Radi vođenja dnevnika prisustva potrebna je programibilna memorija a najjednostavnije i najčešće rješenje je pomoću SD kartice.



**Slika 6. Modul za mikro SD karticu**

Modul služi za čitanje, pisanje i brisanje podataka sa SD kartice pomoću mikroupravljača, a za to koristi SPI komunikaciju. Bitno je napomenuti da SD kartica mora biti u formatu "FAT16" (kartice sa memorijom od 128MB do 2GB) ili "FAT32" (kartice sa memorijom od 4~32GB) jer kartice sa većim memorijama (format "exFAT") nisu kompatibilne sa ovakvim modulima.

### 2.5.1. SPI komunikacija

SPI komunikacija (engl. Serial Peripheral Interface) je vrsta sinkrone "full duplex" komunikacije između glavnog ("master") i podređenog ("slave") uređaja koja se odvija pomoću sljedećih pinova:

- CLK ("Clock") je pin preko kojeg se fizički sinkronizira radni takt slave-a sa radnom frekvencijom master-a.
- CS/SS ("Chip Select" / "Slave Select") je pin preko kojeg master uređaj daje signal podređenom uređaju da želi započeti komunikaciju i da ga taj uređaj mora "slušati".
- MOSI (Master Out-Slave In) je pin preko kojeg nadređeni uređaj šalje podatke prema podređenom uređaju.
- MISO (Master In-Slave Out) je pin preko kojeg nadređeni uređaj prima podatke odaslane od strane podređenog uređaja.

Nakon postavljanja pin-a CS u nisko stanje koje signalizira početak, podaci se šalju u obliku bitova iz posmičnog ("shift") registra master-a u posmični registar slave-a preko pina MOSI i obrnuto preko pina MISO. Full-duplex predstavlja ovu značajku da se istovremeno izmjenjuju podaci iz nadređenog i podređenog uređaja.



## 2.6. RTC sklop

Praćenje vremena je bitan dio zadatka iz razloga što uređaj mora automatski znati koje je predavanje trenutno na rasporedu. Uređaj će to znati tako da će se trenutno vrijeme usporediti sa upisanim rasporedom sati u memoriji. Odabrani RTC je PCF85063A nabavljen sa e-radionice [10].



Slika 7. RTC modul PCF85063A

Radi se o CMOS RTC-u optimiziranom za nisku potrošnju struje. Iz tehničke dokumentacije [11] su izdvojene ove karakteristike čipa:

- praćenje godine, mjeseca, dana, dana u tjednu, sata, minuta i sekundi pomoću 32.768 kHz kristala
- napon napajanja 0.9~5.5V
- potrošnja struje 0.22  $\mu$ A pri  $V_{DD} = 3.3V$  i  $T_{ok} = 25^\circ C$
- funkcija alarma preko interrupt pina

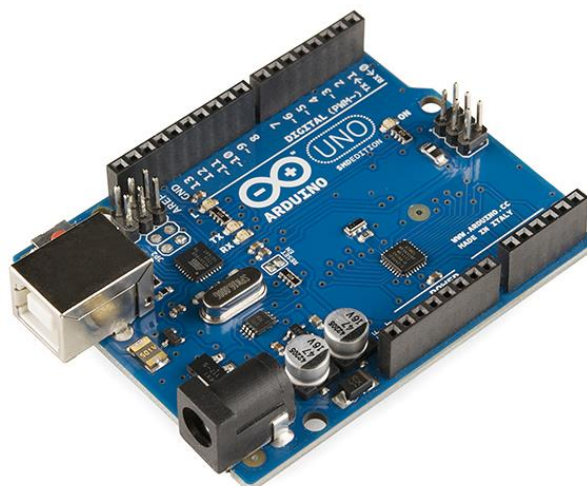
RTC moduli često imaju i utor za bateriju (u ovom slučaju 3V CR2032) kako se ne bi izbrisalo trenutno vrijeme kod gašenja uređaja. Za prijenos podataka o vremenu koristi I2C pinove. Ukoliko se registracija studenata bude izvršavala samo u uskom intervalu vremena (npr. prvih 15 minuta predavanja) pogodno je kalibrirati RTC po proceduri iz dokumentacije [11] i koristiti čip u korekcijskom modu. Tako se postiže preciznost od -2 do 2 ppm sekunde prema [11], što odgovara odstupanju od 0.17 s na dan. Ukoliko je registracija moguća tijekom cijelog predavanja, teoretska nepreciznost vremena od npr. 20ak ppm sekunde će kroz semestar (cca 170 dana) iznositi odstupanje od 5 minuta. U tom slučaju bi bilo pogodno provjeriti odstupanje RTC-ovog vremena svakih mjesec dana kako ne bi došlo do nevažćih prijava izvršenih u točnom, stvarnom vremenu.

## 2.7. Mikroupravljač

Odabrana platforma za programiranje je Arduino IDE. Razlog ovog odabira je izrazito velika zajednica korisnika, jednostavan i besplatan pristup informacijama te najvažnije, podrška za različite sklopove i uređaje u obliku gotovih biblioteka (engl. Libraries) koje uvelike olakšavaju programiranje. Odabrana mikroupravljačka pločica je Arduino UNO iz razloga što su njegove tehničke specifikacije na dovoljno visokoj razini za rješavanje ovog zadatka, ali i zato jer je to najčešće korišteni mikroupravljač za osobnu upotrebu.

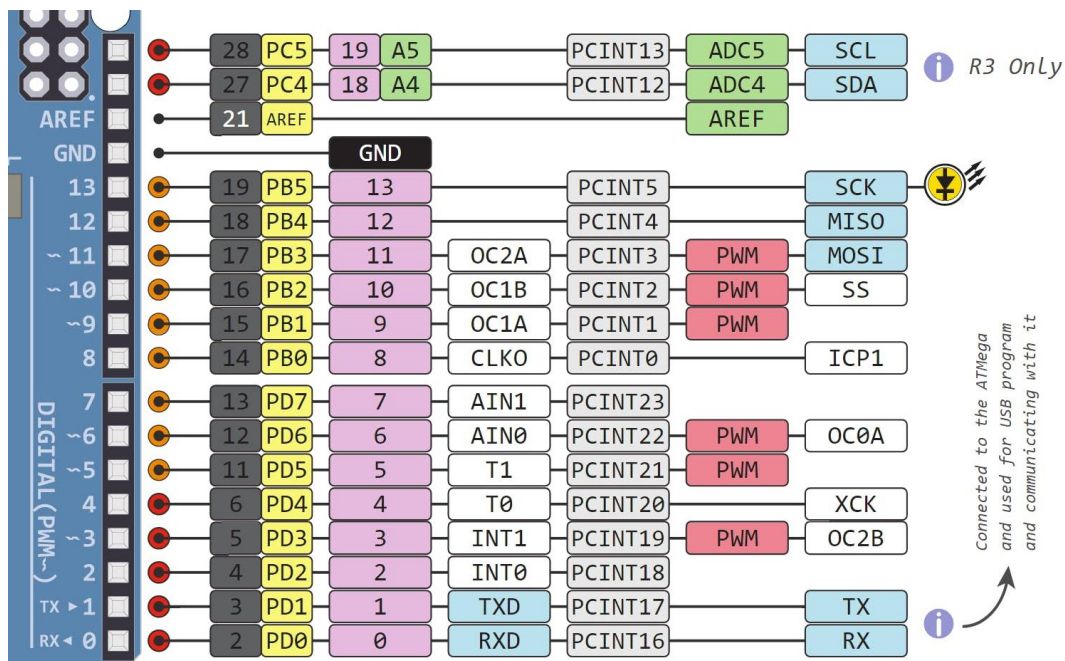
Neke od relevantnih tehničkih specifikacija pločice izvedene iz [12] su:

- čip mikroupravljača je 5V-ni ATmega328P
- 16 MHz oscilator
- napon napajanja ispred stabilizatora 7~12V
- 14 digitalnih I/O pinova, od kojih 6 podržava PWM
- 6 analognih ulaznih pinova
- DC struja po I/O pinu = 20 mA
- DC struja iz 3.3V pina = 50 mA
- 32 KB programibilne memorije (0.5 KB zauzima bootloader)
- 2 KB SRAM memorije
- 1 KB EEPROM memorije



Slika 8. Arduino UNO

Za uređaj je bitno da mikroupravljač podržava I2C, SPI i UART komunikaciju.



Slika 9. Pregled pinova UNO pločice, preuzeto sa [13]

Na slici 9 su vidljivi ti pinovi:

- SDA i SCL (18 i 19) za I2C
- MOSI, MISO i SCK (11, 12 i 13) za SPI
- RX i TX (0 i 1) za UART

## 2.8. Bluetooth komunikacija

Bluetooth komunikacija između uređaja i mobitela ili laptopa će se odvijati pomoću HC-05 Bluetooth modula. Modul se napaja sa 5V a pinovi za UART komunikaciju (RXD i TXD) dugoročno podržavaju samo 3.3V-nu logiku. To znači da je na ulaznom pinu (RXD) potrebno naponsko dijelilo koje će smanjiti izlazni napon iz mikroupravljača na prihvatljivu razinu (par 1-2 k $\Omega$ ). Ovi moduli su tvornički postavljeni da rade sa 'Baud rate'-om od 9600 (prijenos 9600 bitova u sekundi) što znači da i serijska komunikacija mikroupravljača mora biti tako postavljena. Više informacija može se pronaći u tehničkoj dokumentaciji [14].



Slika 10. HC-05 Bluetooth modul

### 2.8.1. UART komunikacija

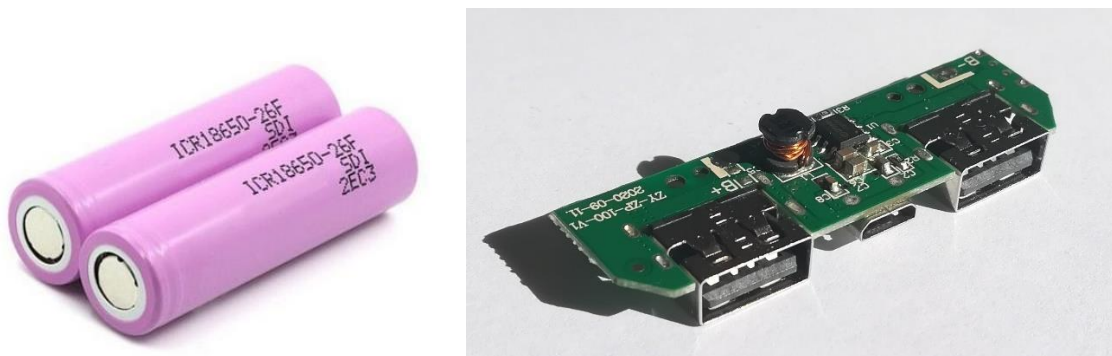
UART (engl. Universal Asynchronus Receiver/Transmitter) je vrsta sinkrone serijske komunikacije sa dva pina:

- RX pin na kojeg dolaze informacije sa drugog uređaja (spojen na TX drugog uređaja)
- TX pin preko kojeg se šalju informacije do drugog uređaja (spojen na RX drugog)

Komunikacija se odvija po sličnom principu kao kod pina SDA kod I2C komunikacije, naizmjeničnim postavljanjem voda na visoki ili niski nivo (logičke nule i jedinice). Bitna razlika je u tome što nema voda za sinkronizaciju radnih taktova uređaja, već se to odvija software-ski na način da se prije poruke šalju sinkronizacijski bitovi. Primatelj broji te bitove kako bi dobio uvid u frekvenciju pošiljateljeve poruke (8 do 16 puta manje frekvencije od uzorkovanja). Primatelj prema tome vremenski usklađuje uzorkovanje/čitanje na polovici svakog bita poruke [15].

## 2.9. Napajanje

Za napajanje su korištene dvije Li-Ion punjive ćelije od 3.7V, spojene u paralelu sa pripadajućim DC-DC konverterom. Konverter služi za podizanje napona na 5V te ima sklop i priključak za punjenje. Navedene komponente su izvađene iz prijenosnog punjača mobitela [16].

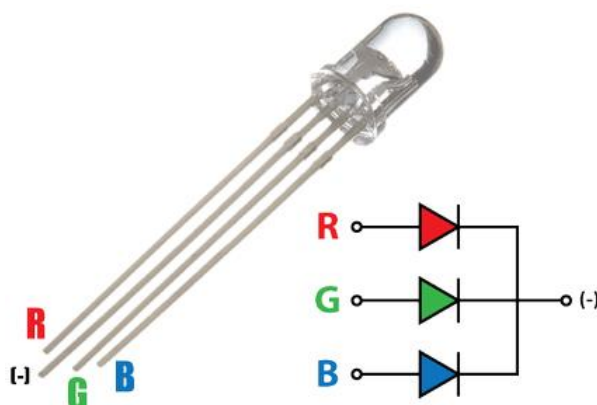


Slika 11. Ćelije i 3.7-5V DC-DC konverter

Navodni nazivni kapacitet ćelija je 2000 mAh što rezultira ukupnom kapacitetu od 4 Ah.

## 2.10. LED u boji

U svrhu povratne informacije tj. vizualne potvrde upisa studenata i minimalne potrošnje energije, korištena je 3-bojna RGB dioda. Kao alternativa može se koristiti LCD display, no njegova korisnost je upitna u ovom slučaju. Ukoliko se očekuje ubrzanje procesa registracije, bljesak lampice daje bržu povratnu informaciju od čitanja slova sa displaya, a za upravljanje upisanim podacima ionako će biti korišteno sučelje na mobitelu ili laptopu.



Slika 12. RGB dioda, zajednička katoda

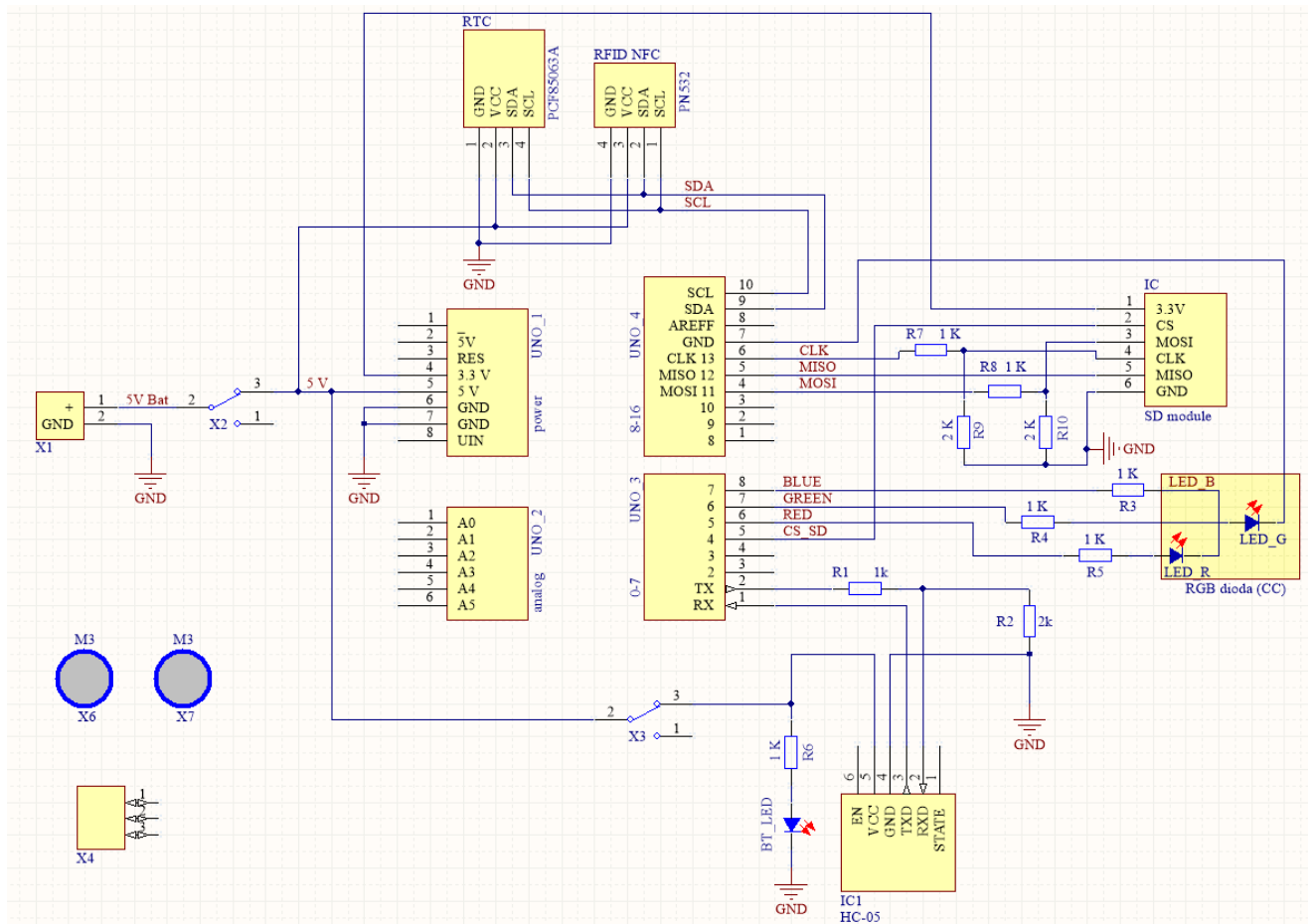
## **2.11. Vizualna potvrda**

Programirani odzivi lampice će biti slijedeći:

1. Kod paljenja uređaja:
  - a. 3 uzastopna zelena blijeska za uspješnu inicijalizaciju SD kartice
  - b. 3 uzastopna crvena blijeska za neuspješnu inicijalizaciju
2. Kod registracije studenata:
  - a. 1 zeleni blijesak ukoliko je student uspješno upisan
  - b. 1 plavi blijesak ukoliko se student pokušao ponovno registrirati
  - c. 2 crvena blijeska ukoliko je došlo do pogreške čitanja ili se registrirao student koji nije u bazi podatka
  - d. 1 crveni blijesak ukoliko trenutno ne traju predavanja
3. Kada je upaljen Bluetooth:
  - a. cijelo vrijeme svijetli plava boja

### 3. Shema pločice uređaja

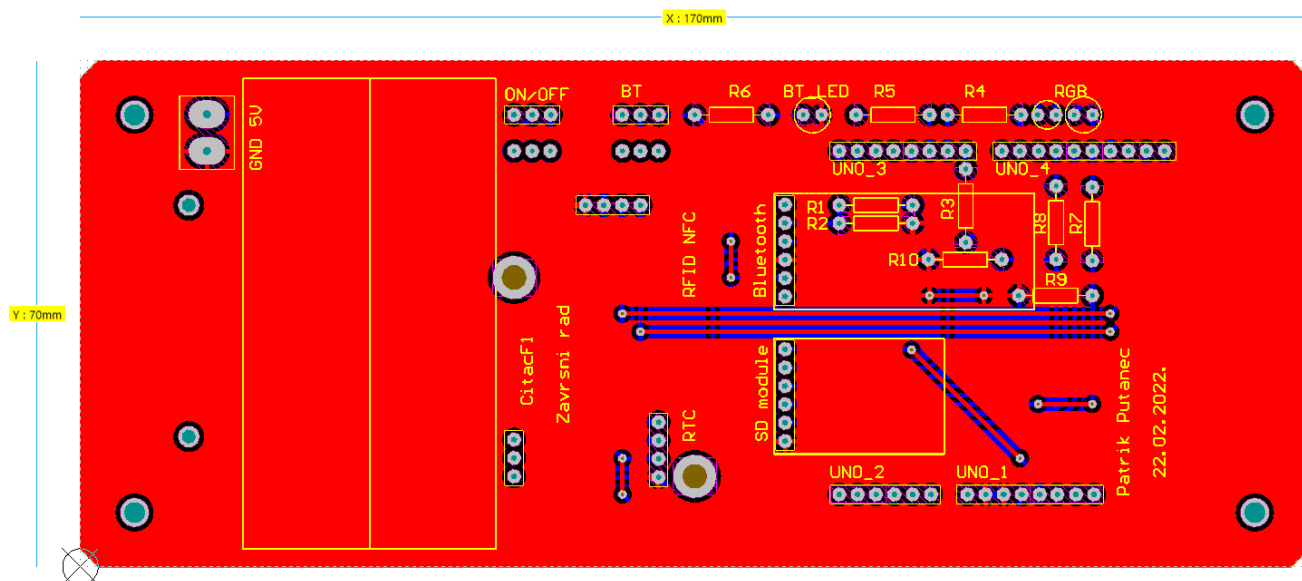
Shema spajanja i model upravljačke PCB pločice napravljeni su uz pomoć programa *Altium Designer* (v17.0).



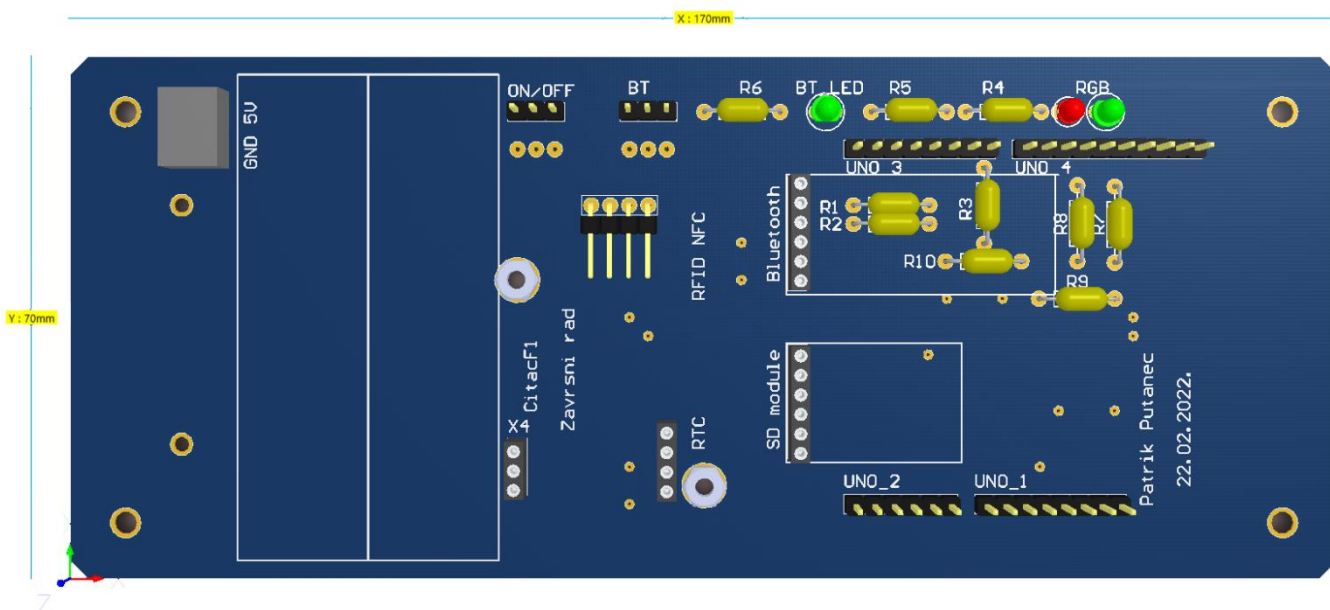
**Slika 13. Električna shema uređaja**

DC-DC konverter iz 7. poglavlja nije prikazan u shemi. Pločica je zamišljena kao svojevrsni 'shield', tj. samo kao adapter između mikroupravljačke pločice Arduino UNO-a i perifernih uređaja. Na pločici su zalemljene komponente koje nisu moduli: otpornici, diode, tipke i konektori. Moduli se na pločicu spajaju na predviđena mjesta preko adekvatnih muško-ženskih konektora, što znači da je njihovo montiranje i skidanje jednostavno a veza dovoljno čvrsta.





Slika 14. PCB pločica

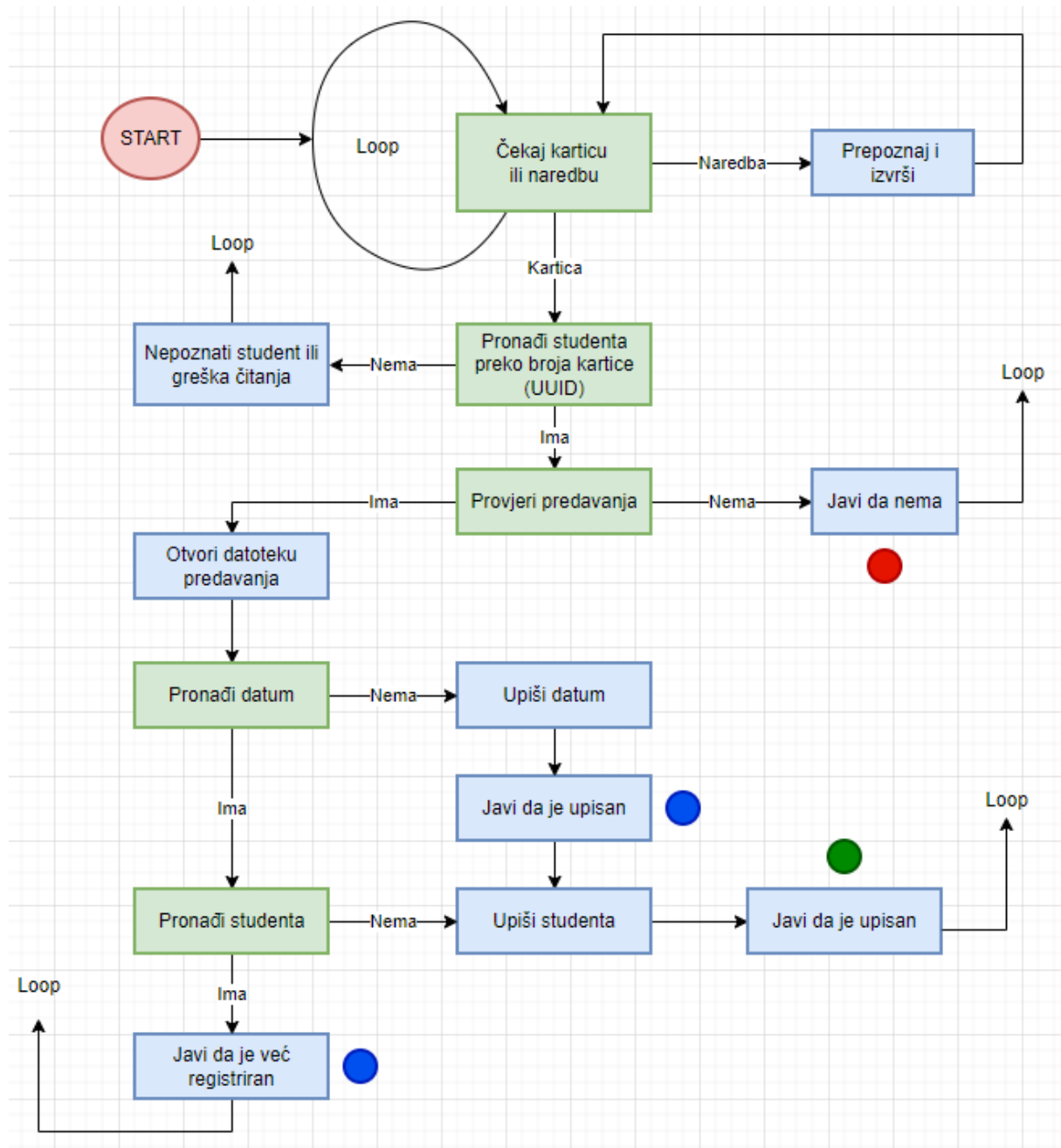


Slika 15. Render PCB pločice sa konektorima i otpornicima



## 4. Idejni program uređaja

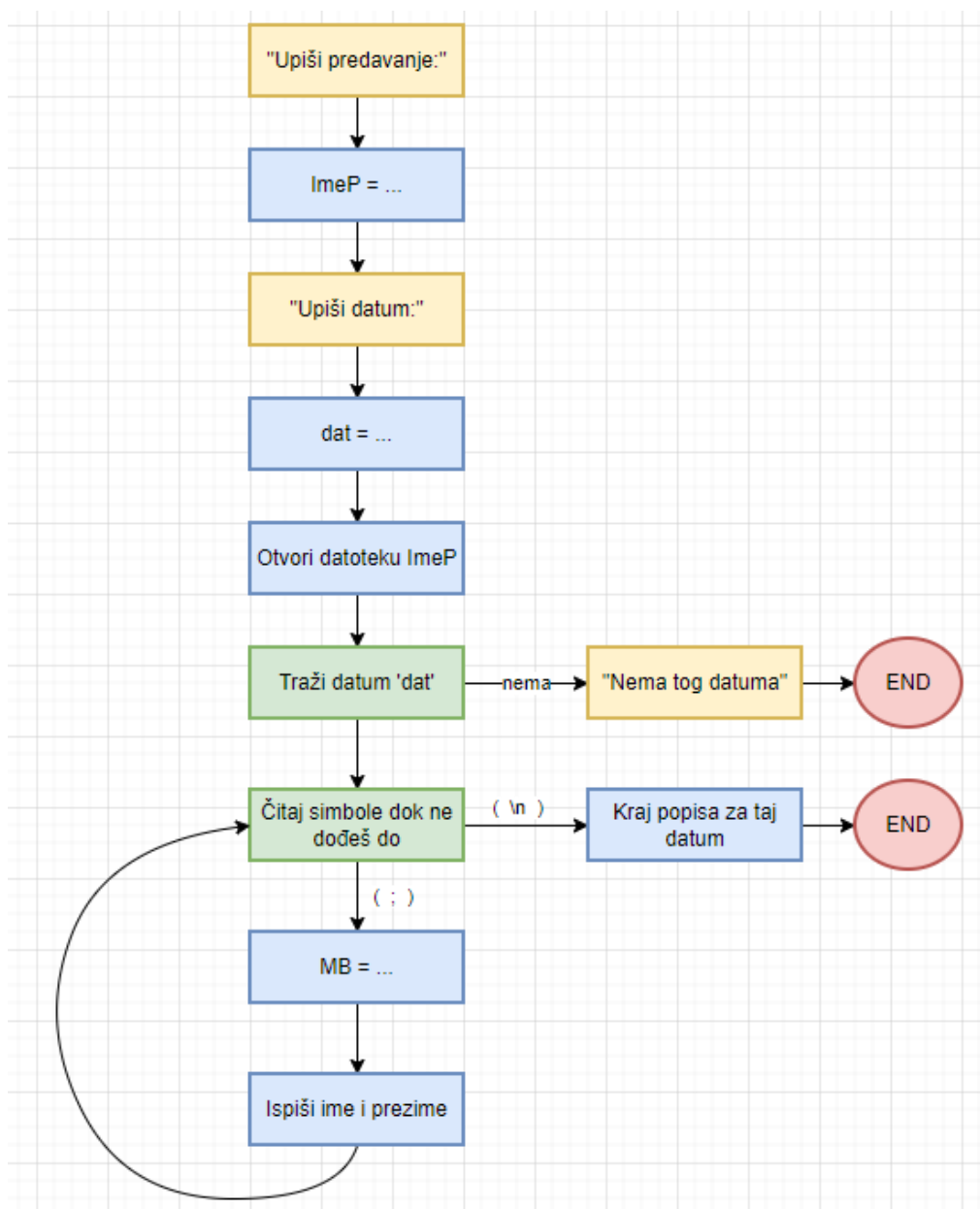
U nastavku je dan idejni tijek programa.



Slika 16. Glavna petlja u programu

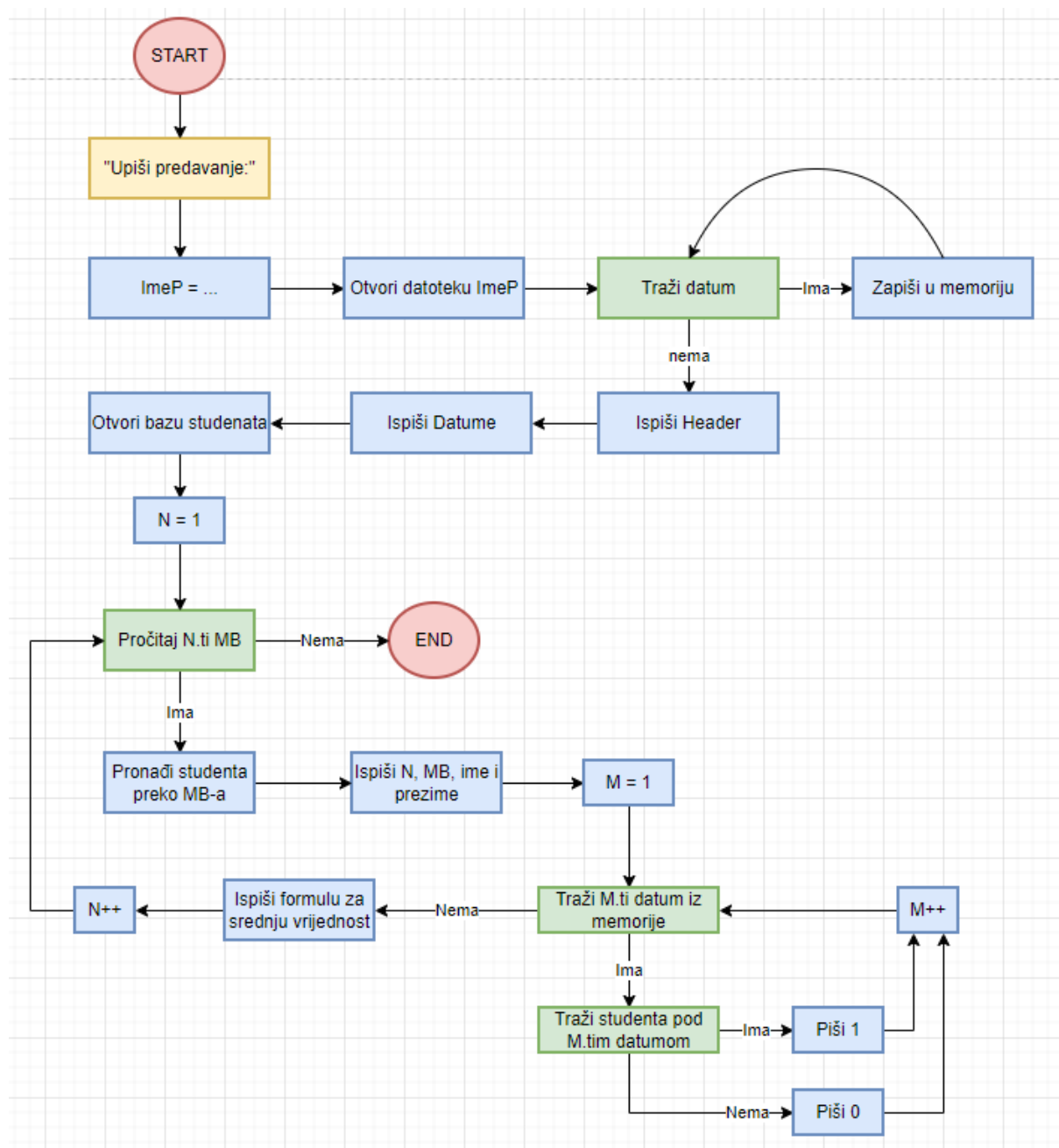
U osnovi, program se sastoji od petlje u kojoj čeka ili naredbu ili očitavanje kartice. Kada mu preko Bluetootha i serijske veze dođe poznata naredba (u obliku riječi), program je prepoznaje kao odvojenu funkciju te se nakon izvršenja vraća u glavnu petlju. Kada se kartica približi beskontaktnom čitaču, on je automatski čita i kao izlaz daje jedinstveni broj te X-ice u formatu od 4 para heksadekatskih vrijednosti, npr. "16 AE C4 18".

Prvo se provjerava da li trenutno uopće ima nekih predavanja/vježbi pomoću RTC-a i datoteke sa rasporedom sati. Ukoliko ima, otvara se datoteka tih predavanja te slijedi traženje današnjeg datuma. Ukoliko ga nema (tek prvi student koji se registrirao), automatski se odvajava od prijašnje liste u dokumentu i zapisuje novi datum u novi red, te matični broj tog studenta u red ispod. Ukoliko je pronađen današnji datum u dokumentu, slijedi traženje matičnog broja pod tim datumom. Ukoliko nema matičnog broja, on se upisuje, a ukoliko je već upisan neće se dogoditi ponovni zapis. Cijeli proces je popraćen vizualnim signalima LED kako je opisano u poglavlju 7.2.



Slika 17. Naredba "NaDan"

Ukoliko se želi provjeriti popis studenata prisutnih na nekom predavanju nekog datuma (iz opisa zadatka 1), za to se koristi Bluetooth komunikacija. Prvo je potrebno uključiti Bluetooth modul pomoću prekidača te se povezati. Nakon uspješnog povezivanja u terminal se upisuje riječ "NaDan" te Enter. Uređaj nam zatim šalje poruku da upišemo ime predavanja te nakon toga, datum. Ukoliko je pronađen željeni datum u datoteci odabranih predavanja, kreće rutina koja automatski čita matične brojeve te ispisuje povezana imena i prezimena. Ova funkcija je prikazana na slici 17. na prethodnoj stranici.



Slika 18. Naredba "Tablica"

Ukoliko se želi dobiti ukupna statistika dolaska svakog studenta (iz opisa zadatka [↑](#)), za to je napravljena funkcija "Tablica". Upisivanjem u terminal, uređaj nas pita na koji predmet (datoteku predavanja) mislimo. Upisivanjem imena predavanja započinje opsežna rutina:

1. Otvara se željena datoteka
2. Pronalaze se svi datumi održavanja unutar te datoteke
3. Ti datumi se upisuju u privremenu datoteku kako ne bi trošili unutarnju memoriju mikroupravljača
4. Ispisuje se 'Header', tj. skup simbola koji odgovara stupcima tablice (Broj, matični broj, Prezime, Ime)
5. Ispisuju se svi datumi po redu (ispisan je prvi red tablice)
6. Otvara se datoteka sa popisom studenata i dohvaća se prvi matični broj
7. Ispisuje se redni broj te podaci uz matični broj: Prezime i Ime
8. Ulazi se u petlju koja traži matični broj studenta pod svim raspoloživim datumima u dokumentu
9. Pod datumom gdje je student pronađen ispisuje se '1' a inače '0'
10. Po rješavanju svih datuma ispisuje se funkcija formatirana za excel tablicu kako bi se dobio postotak prisutnosti za tog studenta (pomoću spomenutih jedinica i nula)
11. Iz datoteke sa popisom studenata se dohvaća slijedeći broj te se ponavljaju radnje od 7. koraka do ovdje
12. Ukoliko više nema matičnih brojeva, to znači da su se ispisale sve vrijednosti za sve studente te se ovdje funkcija završava i vraća u glavnu petlju.

## 5. Program uređaja

Kako je spomenuto u 5. poglavlju, za programiranje je odabran program Arduino IDE. U kôdu su od osnovnih biblioteka dostupnih sa programom korištene:

- **Wire.h** za korištenje I2C komunikacije sa RTC-om i čitačem kartice,
- **SPI.h** za korištenje SPI komunikacije sa SD modulom i
- **SD.h** za pisanje i čitanje sa SD kartice.

A dodatne biblioteke specifične za korištene module su:

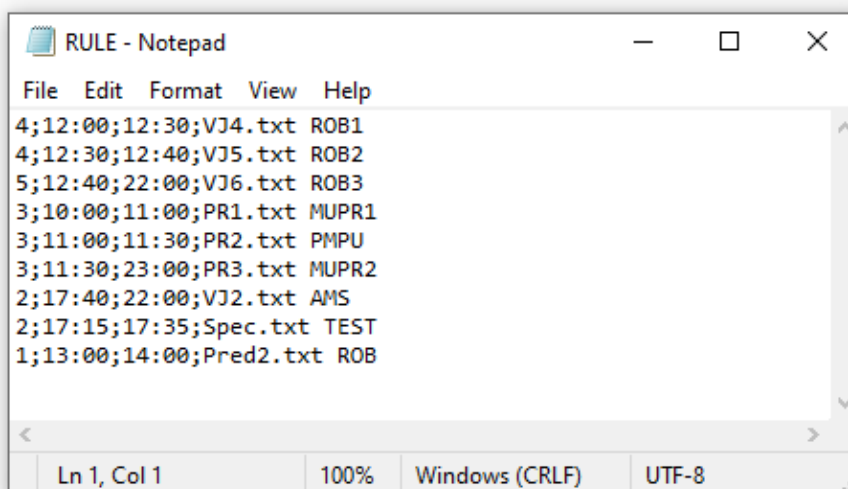
- **PN532\_I2C.h** i **PN532.h** kao podrška za beskontaktni čitač kartice PN532 opisan u poglavlju 2.4. Biblioteka se može pronaći na linku [17].
- Biblioteka **PCF85063A.h** za RTC opisan u poglavlju 4, može se pronaći na linku [18].

Stvarni program je funkcijski vrlo sličan idejnom programu, no jasno da nije tako općenit. Objašnjenje pravog koda ovdje nije napisano ali postoje određene natuknice u kodu koji je dan u prilogu.

U nastavku su prikazane datoteke sa SD kartice koje omogućavaju rad uređaja.

### 5.1. Raspored predavanja

Datoteka je ustvari običan text dokument. Prva koja je prikazana (na slici 19.) je 'RULE.txt' datoteka sa zapisanim rasporedom predavanja. Ovu i datoteku baze studenata je potrebno samostalno napisati i prirediti za upotrebu uređaja.



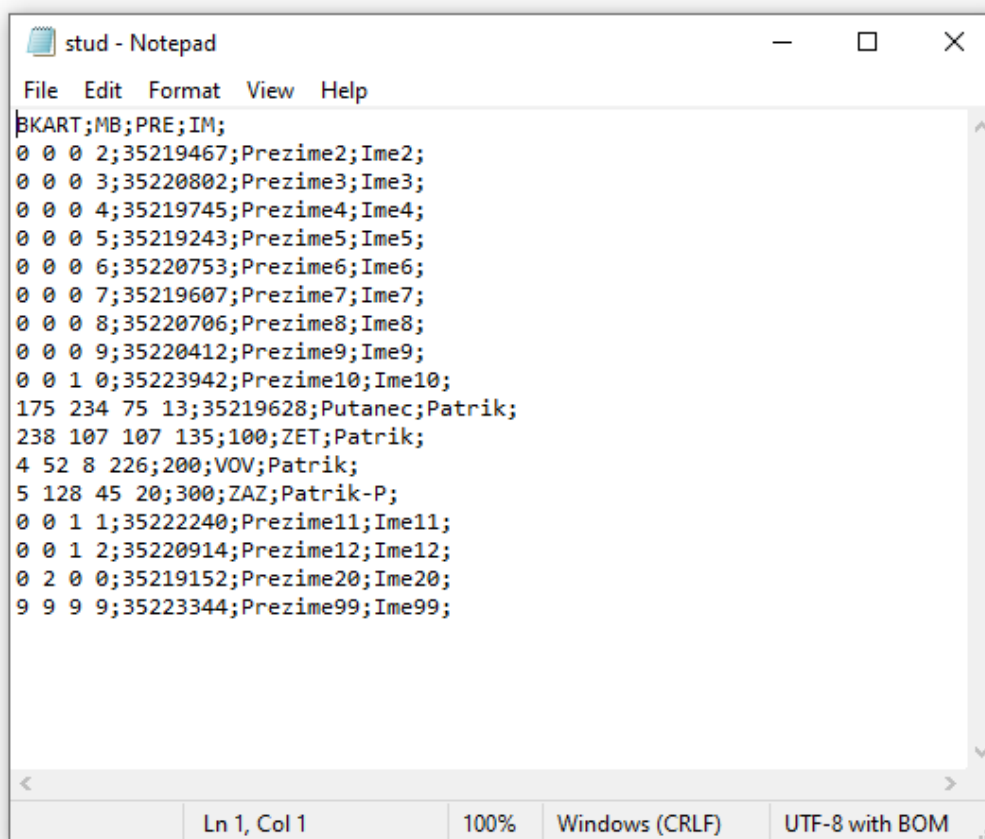
Slika 19. 'RULE.txt' -datoteka sa rasporedom predavanja

Prvi broj koji se upisuje je dan u tjednu na način da 1 odgovara ponedjeljku a 5 -petku. Budući da RTC sadrži vlastiti kalendar za praćenje dana u mjesecu, ova informacija je dovoljna, nije potrebno upisivati svaki datum na koji će se održati to predavanje. Slijedeći brojevi odgovaraju vremenu između kojeg će biti moguća registracija studenata za to predavanje. Tako npr. 12:00;12:30 odgovara dozvoli prijave u intervalu od 15 minuta prije i nakon početka predavanja koje teoretski počinje u 12:15. Alternativno, može se staviti da je prijava moguća tijekom cijelog trajanja predavanja, npr. 12:15;14:45. Važno je držati se zadanog formata gdje su sati i minute odvojene dvotočkom (":") a svaki blok informacija sa točkom-zarez (";") jer je mikroupravljač tako programiran. Nakon intervala trajanja slijedi naziv .txt datoteke u koju će se upisivati dnevnik za to predavanje (npr. VJ4.txt). Na kraju se može staviti razmakom odvojeni komentar ili pravi naziv predavanja ili vježbi koje taj tekst dokument predstavlja.

## 5.2. Baza studenata

Slijedeći dokument je baza studenata koja u slučaju korištenja magnetskog čitača ne mora postojati (kako je objašnjeno u poglavlju 2.3) a u ovom slučaju je ključna. U njoj se nalaze podaci koji će se upisivati u tablicu (matični broj, prezime i ime) a povezani su sa UID-om studentove X-ice. Postavlja se pitanje kako povezati te informacije. Dakle, potrebno je prethodno skenirati karticu kako bi se otkrio njen broj, te tek onda uvesti taj broj u bazu. Glede zaštite osobnih podataka, ovo je najveća mana ovakvog sustava. Potencijalno rješenje je u korištenju RFID tag-ova danih od strane nastavnika no to izlazi iz okvira ove teme...

Znači, prvo je upisan UID sa kartice, matični broj, prezime i ime a sve je međusobno odvojeno točkom-zarez. Slika je na slijedećoj stranici.

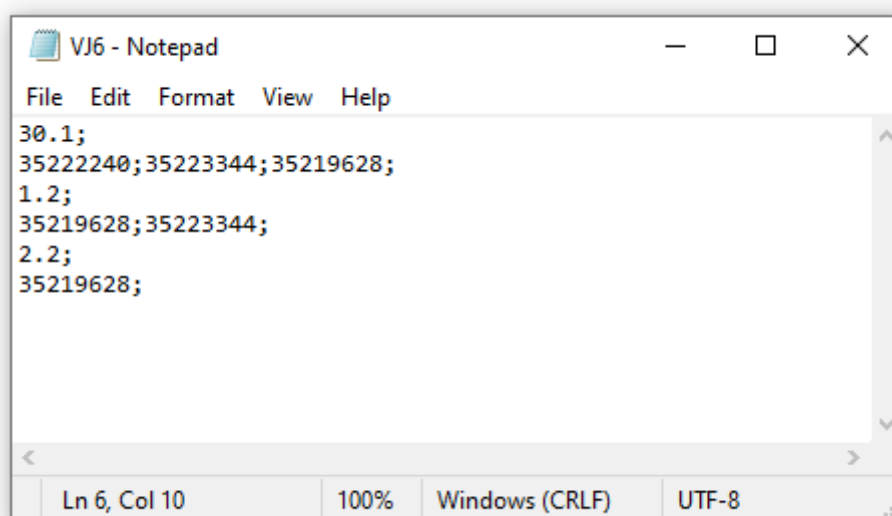


```
stud - Notepad
File Edit Format View Help
BKART;MB;PRE;IM;
0 0 0 2;35219467;Prezime2;Ime2;
0 0 0 3;35220802;Prezime3;Ime3;
0 0 0 4;35219745;Prezime4;Ime4;
0 0 0 5;35219243;Prezime5;Ime5;
0 0 0 6;35220753;Prezime6;Ime6;
0 0 0 7;35219607;Prezime7;Ime7;
0 0 0 8;35220706;Prezime8;Ime8;
0 0 0 9;35220412;Prezime9;Ime9;
0 0 1 0;35223942;Prezime10;Ime10;
175 234 75 13;35219628;Putanec;Patrik;
238 107 107 135;100;ZET;Patrik;
4 52 8 226;200;VOV;Patrik;
5 128 45 20;300;ZAZ;Patrik-P;
0 0 1 1;35222240;Prezime11;Ime11;
0 0 1 2;35220914;Prezime12;Ime12;
0 2 0 0;35219152;Prezime20;Ime20;
9 9 9 9;35223344;Prezime99;Ime99;
```

Slika 20. 'stud.txt' -baza studenata

### 5.3. Primjer datoteke predavanja

Na slici 21 je vidljiv format zapisa studenata na predavanje.



```
VJ6 - Notepad
File Edit Format View Help
30.1;
35222240;35223344;35219628;
1.2;
35219628;35223344;
2.2;
35219628;
```

Slika 21. Primjer datoteke predavanja - 'VJ6.txt'

Ispod svakog datuma su upisani matični brojevi studenata koji su se registrirali, sve informacije su odvojene točkom-zarez (";"). Svaki predmet ima vlastiti tekst dokument te zato nema granice u količini različitih predavanja. Iako mikroupravljač sam može stvarati i brisati dokumente, preporučam da se ručno naprave prazne datoteke za predavanja kako bi ostale u .txt formatu a ne .DAT formatu.

Posljednja datoteka koju je potrebno ručno ubaciti je 'num.txt'. Služi za pretvaranje indeksa stupca u tablici sa brojčane vrijednosti na abecedno slovo (stupci u excel-u su označeni slovima a redci brojevima). Postoji posmak od 4 broja/slova jer se postotak prisutnosti studenta računa samo po stupcima sa datumima (prva 4 stupca su broj, MB, prezime i ime). Ukoliko se očekuje više od 20 različitih datuma po predmetu, potrebno je proširiti num.txt datoteku po pravilima iz excel tablice (nakon Z idu AA, AB, AC itd.).



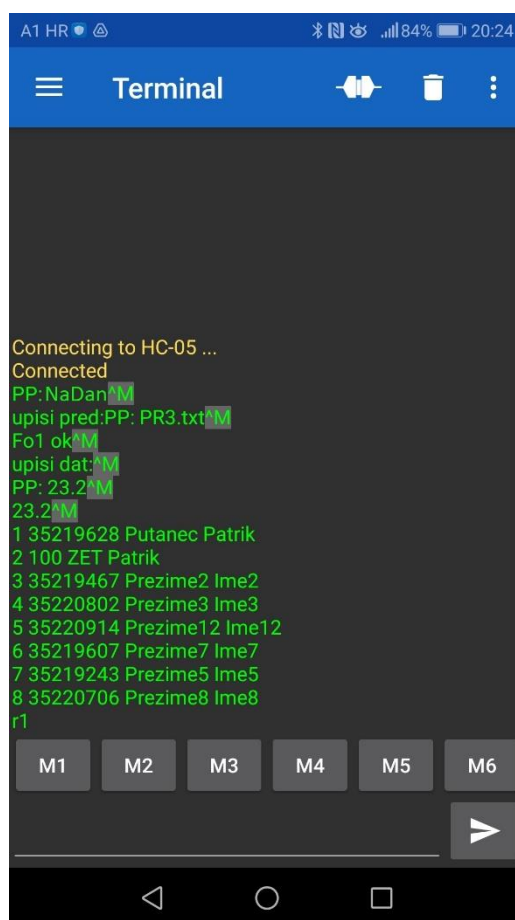
Slika 22. pomoćna 'num.txt' datoteka



#### 5.4. Verifikacija programa

Provedeno je ispitivanje sa bazom od ~20 fiktivnih studenata, te se registracija obavljala preko kartica i ručnim upisivanjem preko terminala. Tako je popunjeno 5 različitih datoteka odnosno predmeta na različite dane i u različite sate.

Na slici 23. je prikazana interakcija sa uređajem uz pomoć mobitela i aplikacije '*Serial Bluetooth Terminal*', dostupne na Google Play-Store-u [19]. Naravno, isti rezultat se može postići i sa bilo kojom drugom Bluetooth Terminal aplikacijom.



Slika 23. Naredba 'NaDan' preko mobitela

Vidljiv je upis naredbe 'NaDan', upis imena datoteke 'PR3.txt' te datum '23.2' nakon čega slijedi popis prisutnih studenata.

Radi jednostavnosti, za sučelje na laptopu je korišten Arduino-ov 'Serial monitor' na način da se laptop-ov bluetooth interno spoji na jedan od COM-portova (npr. COM6) a Baud rate se namjesti na 9600.

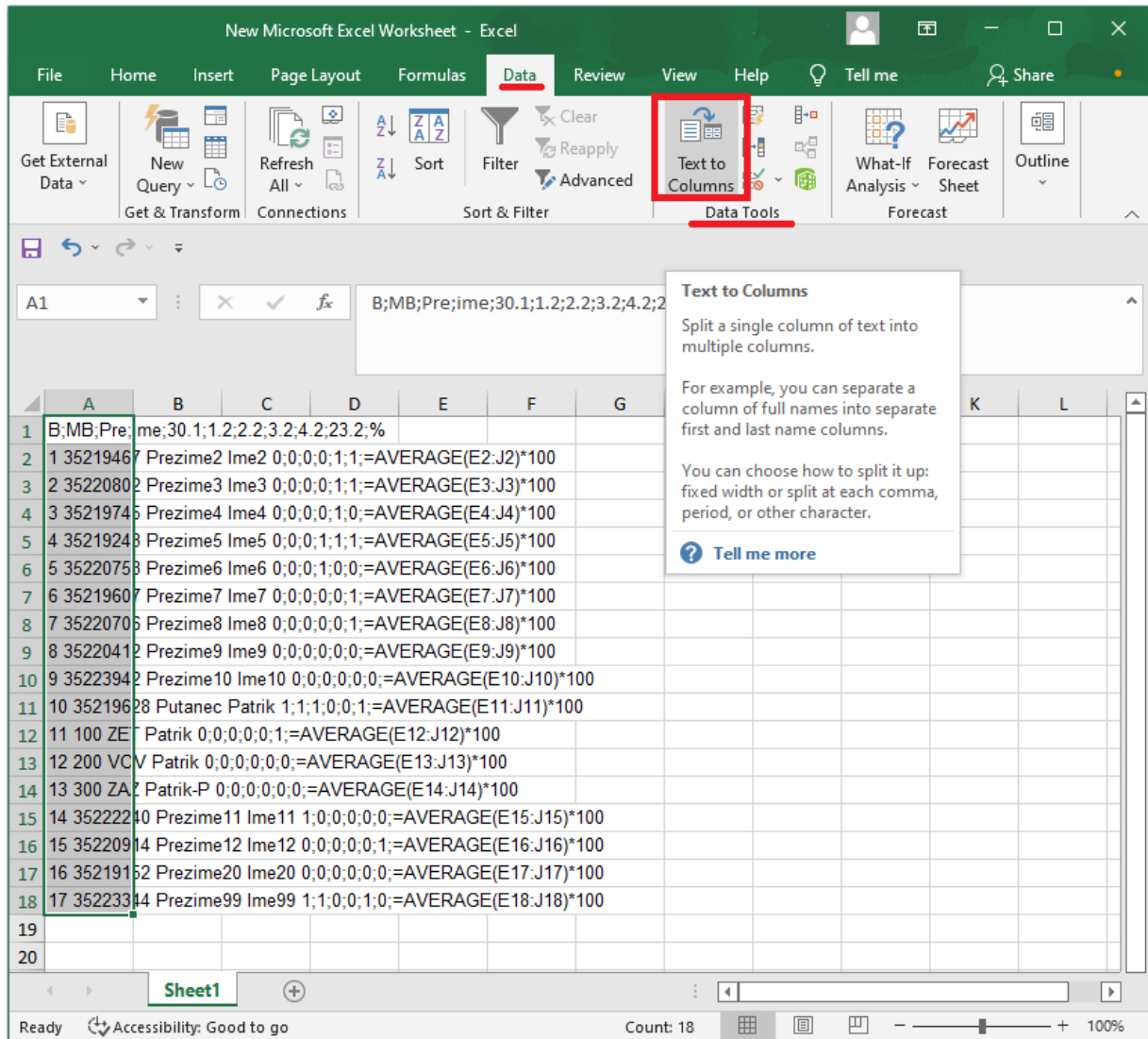
```

COM6
PP: tabl
PP: PR3.txt
Fo1 ok
---
B;MB;Pre;ime;30.1;1.2;2.2;3.2;4.2;23.2;%
1 35219467 Prezime2 Ime2 0;0;0;0;1;1;=AVERAGE (E2:J2) *100
2 35220802 Prezime3 Ime3 0;0;0;0;1;1;=AVERAGE (E3:J3) *100
3 35219745 Prezime4 Ime4 0;0;0;0;1;0;=AVERAGE (E4:J4) *100
4 35219243 Prezime5 Ime5 0;0;0;1;1;1;=AVERAGE (E5:J5) *100
5 35220753 Prezime6 Ime6 0;0;0;1;0;0;=AVERAGE (E6:J6) *100
6 35219607 Prezime7 Ime7 0;0;0;0;0;1;=AVERAGE (E7:J7) *100
7 35220706 Prezime8 Ime8 0;0;0;0;0;1;=AVERAGE (E8:J8) *100
8 35220412 Prezime9 Ime9 0;0;0;0;0;0;=AVERAGE (E9:J9) *100
9 35223942 Prezime10 Ime10 0;0;0;0;0;0;=AVERAGE (E10:J10) *100
10 35219628 Putanec Patrik 1;1;1;0;0;1;=AVERAGE (E11:J11) *100
11 100 ZET Patrik 0;0;0;0;0;1;=AVERAGE (E12:J12) *100
12 200 VOV Patrik 0;0;0;0;0;0;=AVERAGE (E13:J13) *100
13 300 ZAZ Patrik-P 0;0;0;0;0;0;=AVERAGE (E14:J14) *100
14 35222240 Prezime11 Ime11 1;0;0;0;0;0;=AVERAGE (E15:J15) *100
15 35220914 Prezime12 Ime12 0;0;0;0;0;1;=AVERAGE (E16:J16) *100
16 35219152 Prezime20 Ime20 0;0;0;0;0;0;=AVERAGE (E17:J17) *100
17 35223344 Prezime99 Ime99 1;1;0;0;1;0;=AVERAGE (E18:J18) *100
18 end
  
```

Slika 24. Naredba 'tabl' preko laptopa

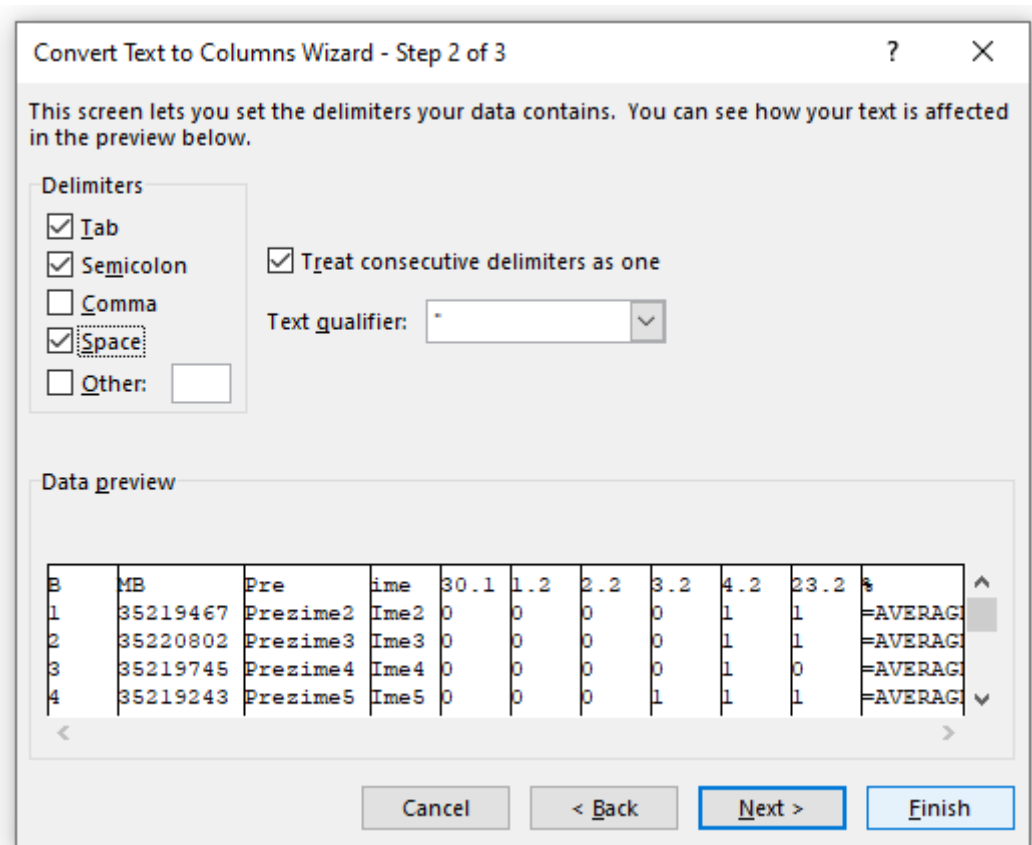
Na slici 24. je vidljiva interakcija sa uređajem uz pomoć laptopa i Arduino-vog serial monitora. Vidljiv je upis naredbe za tablicu 'tabl', upis imena datoteke predavanja 'PR3.txt' te generirana tablica podataka. Generiranje tablice je najzahtjevniji dio programa, ukoliko je upisano puno studenata i datuma proces može potrajati par sekundi.

Unutar monitora se pomoću miša označe svi podaci između zadnjeg reda i tri crtice ('---') te se pomoću tipkovnice kopiraju (Ctrl + C). U otvorenom excel dokumentu označimo prvu ćeliju (A1) te unesemo podatke (Ctrl + V).



Slika 25. Ubacivanje podataka u tablicu

Kako je vidljivo sa slike 25, excel automatski prepoznaje novi red te dijeli podatke na redove. Još je samo potrebno podijeliti podatke na stupce. Odabirom kartice 'Data' i naredbe 'Text to Columns' otvara se izbornik u kojem odabiremo 'Delimited' i 'Next'. U sljedećem prozoru (Slika 26) odabiremo 'Semicolon' i 'Space'. Klikom na 'Finish' konačno dobivamo gotovu tablicu evidencije za to predavanje (primjer na slici 27).



Slika 26. Razdvajanje podataka na stupce

	A	B	C	D	E	F	G	H	I	J	K	L
1	B	MB	Pre	ime	30.sij	1.vlj	2.vlj	3.vlj	4.vlj	23.vlj	%	
2		1 35219467	Prezime2	Ime2	0	0	0	0	1	1	33,33333	
3		2 35220802	Prezime3	Ime3	0	0	0	0	1	1	33,33333	
4		3 35219745	Prezime4	Ime4	0	0	0	0	1	0	16,66667	
5		4 35219243	Prezime5	Ime5	0	0	0	1	1	1	50	
6		5 35220753	Prezime6	Ime6	0	0	0	1	0	0	16,66667	
7		6 35219607	Prezime7	Ime7	0	0	0	0	0	1	16,66667	
8		7 35220706	Prezime8	Ime8	0	0	0	0	0	1	16,66667	
9		8 35220412	Prezime9	Ime9	0	0	0	0	0	0	0	
10		9 35223942	Prezime10	Ime10	0	0	0	0	0	0	0	
11		10 35219628	Putanec	Patrik	1	1	1	0	0	1	66,66667	
12		11 100	ZET	Patrik	0	0	0	0	0	1	16,66667	
13		12 200	VOV	Patrik	0	0	0	0	0	0	0	
14		13 300	ZAZ	Patrik-P	0	0	0	0	0	0	0	
15		14 35222240	Prezime11	Ime11	1	0	0	0	0	0	16,66667	
16		15 35220914	Prezime12	Ime12	0	0	0	0	0	1	16,66667	
17		16 35219152	Prezime20	Ime20	0	0	0	0	0	0	0	
18		17 35223344	Prezime99	Ime99	1	1	0	0	1	0	50	
19												

Slika 27. Tablica evidencije za jedan predmet.

## 6. Procjena vrijednosti uređaja

Komponenta	Ime/tip/proizvođač	Količina	Jedinična cijena [HRK]	Ukupna cijena [HRK]
mikrokontroler	Arduino UNO	1	180,00	180,00
mikro SD modul	protis.hr	1	16,00	16,00
mikro SD kartica	16 GB/32 GB	1	50,00	50,00
Bluetooth modul	HC-05	1	107,00	107,00
RTC	PCF85063A	1	19,00	19,00
baterija za RTC	3V CR2032	1	10,00	10,00
bekontaktni čitač	elechouse PN532	1	170,00	170,00
baterija za napajanje	Powerbank PEPCO	1	35,00	35,00
prekidač/sklopka		2	10,00	20,00
RGB dioda	5mm, zajednička katoda	1	1,00	1,00
plava dioda		1	0.20	0.20
otpornik	470 $\Omega$	3	0.30	0.90
	1 k $\Omega$	4	0.30	1.20
	2 k $\Omega$	3	0.30	0.90
konektori-ženski	letva 1x32	1	15,00	15,00
konektor-muški-kutni	letva 1x40	1	2,00	2,00
konektori-muški	letva 1x36	1	7,80	7,80
odstojnik	M3 x 20mm Ž-Ž	1	4,00	4,00
	M3 x 12mm Ž-Ž	1	3,00	3,00
	M3 x 12mm M-Ž	1	4,00	4,00
vijak	M3 6mm	4	1,00	4,00
pločica		1	150,00	150,00
				$\Sigma=801,00$ HRK

**Tablica 1. Ukupni trošak za komponente**

## 7. ZAKLJUČAK

U ovom radu je prikazan odabir komponenti i tijek programa za programiranje uređaja za registraciju studenata preko X-ice. Projektirani uređaj ispunjava sve zahtjeve dane u zadatku i ispitivanjem funkcija je utvrđeno da radi vrlo pouzdano. Komunikacija preko odabranih bluetooth sučelja je jednostavna i učinkovita.

Ukupni utrošak programske memorije je 78% odnosno 25182 / 32256 byta a utrošak radne memorije je 68% (1400 / 2048 byta) što znači da je napravljen dobar odabir mikroupravljača.

Na žalost, iskoristivost cijelog sustava je vrlo upitna glede razloga o zaštiti privatnosti i prikupljanju osobnih podataka sa studentskih kartica. Svakako bi trebalo proučiti zakon o zaštiti osobnih podataka [20].

Ukoliko bi došlo do promjena koje bi omogućile korištenje ovakvog sustava, za prijelaz sa prototipa na ozbiljni uređaj korisno bi bilo izvršiti neke od slijedećih radnji:

- Optimizacija koda sa ciljem stvaranja mjesta u memoriji za dodatne funkcije i/ili programiranje u drugom programu.
- Provjera iznosa vrijednosti pokazivača (pointera) za velike file-ove (umjesto 'int' koristiti 'long' ili čak 'unsigned long') kako ne bi došlo do greški kod traženja ili upisivanja.
- Provjera brzine traženja matičnog broja pri većoj bazi studenata (npr. pri nekoliko stotina).
- Provjera brzine traženja datuma i studenta u datotekama velikih predavanja (npr. dvaput tjedno po 100 studenata tijekom semestra  $\approx$  30000 znakova).
- Revizija i smanjenje PCB pločice projektiranjem vlastitog mikroupravljačkog sklopa.
- Napraviti izračun ukupne potrošnje električne energije uređaja.

## LITERATURA

- [1] <https://pdfslide.net/reader/f/a-day-in-the-life-of-a-flux-reversal>, 20.2.2022.
- [2] [https://en.wikipedia.org/wiki/Magnetic\\_stripe\\_card#Production](https://en.wikipedia.org/wiki/Magnetic_stripe_card#Production), 20.2.2022.
- [3] [https://en.wikipedia.org/wiki/Proximity\\_card](https://en.wikipedia.org/wiki/Proximity_card), 20.2.2022.
- [4] [https://en.wikipedia.org/wiki/Radio-frequency\\_identification](https://en.wikipedia.org/wiki/Radio-frequency_identification), 20.2.2022.
- [5] [https://en.wikipedia.org/wiki/Near-field\\_communication](https://en.wikipedia.org/wiki/Near-field_communication), 20.2.2022.
- [6] <https://www.electronics-notes.com/articles/connectivity/nfc-near-field-communication/tags-types.php>, 20.2.2022.
- [7] [https://www.elehouse.com/elehouse/index.php?main\\_page=product\\_info&cPath=90\\_93&products\\_id=2242](https://www.elehouse.com/elehouse/index.php?main_page=product_info&cPath=90_93&products_id=2242), 21.2.2022.
- [8] [http://www.elehouse.com/elehouse/images/product/PN532\\_module\\_V3/PN532\\_%20Manual\\_V3.pdf](http://www.elehouse.com/elehouse/images/product/PN532_module_V3/PN532_%20Manual_V3.pdf), 21.2.2022.
- [9] <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2c-bus/>, 21.2.2022.
- [10] <https://e-radionica.com/>, 22.2.2022.
- [11] <https://www.nxp.com/docs/en/data-sheet/PCF85063A.pdf>, 22.2.2022.
- [12] [https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf?\\_gl=1\\*9u2lwz\\*\\_ga\\*MjUwMjAxMTk1LjE2MjUxMzZMzZg.\\*\\_ga\\_NEXN8H46L5\\*MTY0NTU1NjY2OS4xMTAuMS4xNjQ1NTU2Njc2LjA](https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf?_gl=1*9u2lwz*_ga*MjUwMjAxMTk1LjE2MjUxMzZMzZg.*_ga_NEXN8H46L5*MTY0NTU1NjY2OS4xMTAuMS4xNjQ1NTU2Njc2LjA), 22.2.2022.
- [13] <https://www.circuito.io/blog/arduino-uno-pinout/>, 22.2.2022.
- [14] [https://www.rcscomponents.kiev.ua/datasheets/hc\\_hc-05-user-instructions-bluetooth.pdf](https://www.rcscomponents.kiev.ua/datasheets/hc_hc-05-user-instructions-bluetooth.pdf), 23.2.2022.
- [15] <https://www.allaboutcircuits.com/technical-articles/back-to-basics-the-universal-asynchronous-receiver-transmitter-uart/>, 23.2.2022.
- [16] <https://www.morele.net/powerbank-pepco-power-bank-pojemnosc-nominalna-6000mah-pattern-bialy-7188038/>, 23.2.2022.
- [17] <https://github.com/elehouse/PN532>, 23.2.2022.
- [18] <https://github.com/e-radionicacom/PCF85063A-Arduino-Library>, 23.2.2022.
- [19] [https://play.google.com/store/apps/details?id=de.kai\\_morich.serial\\_bluetooth\\_terminal&hl=hr&gl=US](https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=hr&gl=US), 23.2.2022.
- [20] [https://narodne-novine.nn.hr/clanci/sluzbeni/2012\\_09\\_106\\_2300.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2012_09_106_2300.html), 24.2.2022.

## **PRILOZI**

- I. CD-R disk
- II. Arduino kôd
- III. Električna shema i PCB pločica



## PRILOG II

### Aduino kôd

```
/* -----  
   Citac kartice VFF - Završni rad  
  
   Patrik Putanec  
   0035 219628  
  
   22.02.2022.  
-----  
*/  
  
//#include <stdint.h>  
//#include <stdbool.h>  
  

```

```

T = 0;
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);

Serial.begin(9600);
SPI.begin();           // Init SPI bus

  debugln("C ON:"); debug(rtc.getDay()); debug('/'); debug(rtc.getMonth());
  debug('/');
  debug(rtc.getYear()); debug(' ');
  Serial.print(rtc.getHour(), DEC); Serial.print(':');
  Serial.println(rtc.getMinute(), DEC);

  SSD();           // Setup for the SD card

  nfc.begin();

  nfc.SAMConfig(); // Configure board to read RFID tags

}

// S S D
void SSD(void) { // funkcija za inicijalizaciju SD kartice
  Serial.print(F("Init SD...")); // ukoliko nije inicijalizirana provjeriti
  if (!SD.begin(CS_SD)) { // da li je "kliknula"?
    Serial.println(F("init failed!"));
    Flash(RED, 3);
    return;
  }
  Serial.println(F("init done."));
  Flash(GREEN, 3);
}

// F L A S H
void Flash(int LED, int times) { // funkcija za blinkanje LED-ica
  for (int i = 1 ; i <= times ; i++) {
    digitalWrite(LED, HIGH);
    delay(330);
    digitalWrite(LED, LOW);
    if (i != times) delay(330);
  }
}

// C I T A J P
void citajP(void) { // funkcija za čitanje Stringova sa Serial
  Monitora
  String izlaz; // testira dobivenu poruku
  delay(10);
  while (Serial.available() > 0) {
    char inChar = Serial.read();
    if (inChar == '\n') {
      Serial.print(F("PP: "));
      debugln(izlaz);
      break;
    }
    izlaz += (char)inChar;
  }
  if (izlaz != "") {
    test(izlaz);
  }
}

// K A R T I C A
bool kartica(int A, String ime) { // funkcija za otvaranje datoteke sa SD
  kartice u File za pisanje i čitanje
  if (A == 1 || A == 2) { // za A = 1 -> openW myFile za A = 2 ->
    openW myFile2
    digitalWrite(CS_SD, LOW); // za A = 0 -> close myFile za A = 3 ->
    close myFile2

```

```

delay(10);
if (A == 1) {
  myFile = SD.open(ime, FILE_WRITE);
  if (myFile) {
    Serial.println(F("Fo1 ok"));
    myFile.seek(0);
    return true;
  }
  else {
    Serial.print(F("errop ")); Serial.println(ime);
    digitalWrite(CS_SD, HIGH);
    return false;
  }
}
if (A == 2) {
  myFile2 = SD.open(ime, FILE_WRITE);
  if (myFile2) {
    //Serial.println(F("Fo2 ok"));
    myFile2.seek(0);
    return true;
  }
  else {
    Serial.print(F("errop ")); Serial.println(ime);
    digitalWrite(CS_SD, HIGH);
    return false;
  }
}
}
else if (A == 0 || A == 3) {
  if (A == 0) {
    myFile.close();
    Serial.println(F("Fcl"));
  }
  if (A == 3) {
    myFile2.close();
  }
  digitalWrite(CS_SD, HIGH);
  return true;
}
}

// S T U P A C
String stupac(int i) {
  stupaca u excel tablici // funkcija za previlno numeriranje
  String stup; // otvara datoteku num.txt u kojoj je za
svaki stupac dodijeljeno veliko // slovo sa offsetom od 4, tako je 1 -> E
  delay(10); // slovo sa offsetom od 4, tako je 1 -> E
, 2 -> F , 3 -> G itd.
  helpF = SD.open("num.txt", FILE_READ); // otvara nezavisni File helpF kako bi
mogli istodobno biti otvereni drugi
  if (helpF) {
    while (helpF.available()) {
PONOVO:
      stup = helpF.readStringUntil(' ');
      if (stup.toInt() == i) {
        stup = "";
        stup = helpF.readStringUntil('\n');
        break;
      }
      else {
        stup = helpF.readStringUntil('\n');
        stup = "";
        goto PONOVO;
      }
    }
  }
  helpF.close();
  stup.trim();
  return stup;
}

```

```

    }
    else {
        Serial.print(F("errorS "));
        return "E";
    }
}

//      T R A Z I
// funkcija za pretraživanje
bool trazi(String ulaz, int A) {
    // broja kartice (A = 0) ili
    // ukoliko je A > 0, traži
    // uz pomoć funkcije kartica()
    // File "myFile2" kako ne bi
    // narušio pohranjeni "myFile"
    base studenata "stud.txt" uz pomoć
    int i = 0;
    preko matičnog broja (A = -1)
    int poz = 0;
    A.tog studenta po redu
    if (A > 0) A++;
    otvara se u File "myFile2"
    myFile2.seek(0);
    while (myFile2.available()) {
        PONOV:
        if (A != 0) { // pretražujemo po MB -> preskoči broj karice
            while (myFile2.peek() != ';') {
                if (myFile2.position() != myFile2.size()) myFile2.seek(myFile2.position() +
1);
                else {
                    kartica(3, studF);
                    return false;
                }
            }
            myFile2.seek(myFile2.position() + 1);
        }
        String tuid = myFile2.readStringUntil(';');
        if (i == 0) {
            i++;
        }
        else {
            tuid.trim();
            i++;
        }
        if (A == 1 || 2 * A - 1 == i) goto IZLAZ;
        poz = myFile2.position();
        if (tuid == "") {
            Serial.println(F("nema"));
            Flash(RED, 3);
            return false;
        }
        else if (tuid.equals(ulaz)) {
            if (A == 0) tuid = myFile2.readStringUntil(';');
        }
        IZLAZ:
        if (A >= 0) MBS = tuid;
        String pre = myFile2.readStringUntil(';');
        String im = myFile2.readStringUntil(';');
        Serial.print(tuid); Serial.print(' '); Serial.print(pre); Serial.print(' ');
Serial.print(im);
        if (A <= 0) Serial.print('\n');
        return true;
    }
    else {
        myFile2.seek(poz);
        tuid = "";
        goto PONOV;
    }
}
}

//      T E S T
// funkcija za grananje programa po
želji
void test(String ulaz) {

```

```

    if (ulaz == "a") { // za ove ulaze postoje slijedeće
funcije:
        T = 1; // a | tražim studenta preko
UID
        debugln("U UID"); // a1 | tražim studenta preko
MB
    } // b | otvori željenu
    datoteku u File "myFile"
    else if (ulaz == "a1") { // c | zatvori željenu
    datoteku
        T = 2; // d | ispiši koja su sad
predavanja
        debugln("U MB"); // reset | resetiraj SD
    } // text | pohrani ime i otvori
    željenu datoteku
    else if (T == 1) { // dump | ispiši sadržaj
    datoteke u File-u "myFile"
        imeF = pred(); // trazi | trazi n.tog studenta
    po redu
        Serial.println(imeF);
        if (imeF != "") {
            if (kartica(2, studF)) {
                Serial.print(F("Traž: "));
                Serial.println(ulaz);
                if (trazi(ulaz, 0)) upisi();
            }
            kartica(3, studF);
        }
        T = 0;
    }
    else if (T == 2) {
        if (imeF != "") {
            if (kartica(2, studF)) {
                Serial.print(F("Tražim: "));
                Serial.println(ulaz);
                trazi(ulaz, -1);
            }
            kartica(3, studF);
        }
        else {
            Flash(RED, 2);
        }
        T = 0;
    }
    else if (ulaz == "b") {
        kartica(1, imeF);
    }
    else if (ulaz == "c") {
        kartica(0, imeF);
    }
    else if (ulaz == "d") {
        imeF = pred();
        Serial.println(imeF);
    }
    else if (ulaz == "reset") {
        SSD();
    }
    else if (ulaz == "text") {
        Serial.print(F("Ime doc->"));
        T = 3;
    }
    else if (T == 3) {
        Serial.print(F("Otv: ")); Serial.println(ulaz);
        T = 0;
        imeF = ulaz;
        kartica(1, imeF);
        dump();
    }

```

```

    }
    else if (ulaz == "NaDan") {
        Serial.print(F("upisi pred:"));
        T = 5;
    }
    else if (T == 5) {
        imeF = ulaz;
        if (kartica(1, imeF)) {
            if (kartica(2, studF)) {
                T = 6;
                Serial.println(F("upisi dat:"));
            }
        }
    }
    else if (T == 6) {
        T = 0;
        ispisiD(ulaz);
    }
    else if (ulaz == "tabl") {
        T = 7; Serial.print(F("upisi pred:"));
    }
    else if (T == 7) {
        T = 0;
        imeF = ulaz;
        Datumi(imeF);
        delay(20);
        Tablica(imeF);
    }
}

String pred() {
    predavanja // P R E D
                // funkcija za određivanje aktualnih
    if (kartica(2, "RULE.txt")) { // otvara datoteku sa pravilima
        predavanja "RULE.txt" u
        int Ts; // provjerava interno vrijeme RTC-a sa
        intervalima predavanja // i vraća ime dokumenta predavanja
        int Te;
        ukoliko trenutno traju
        int Tt;
        // int poz = 0;
        while (myFile2.available()) {
        PONOVI:
            String Dan = myFile2.readStringUntil(';');
            if (Dan == "") {
                Serial.println(F("Sada nema predavanja!"));
                kartica(3, "RULE.txt");
                return "";
            }
            else if (Dan.toInt() == rtc.getWeekday()) { //poklapa se dan u tjednu
                Ts = ((myFile2.readStringUntil(':')).toInt()) * 100 +
(myFile2.readStringUntil(';')).toInt();
                Te = ((myFile2.readStringUntil(':')).toInt()) * 100 +
(myFile2.readStringUntil(';')).toInt();
                Tt = rtc.getHour() * 100 + rtc.getMinute();
                if (Tt >= Ts && Tt <= Te) { //poklapa se i vrijeme
                    String Pred = myFile2.readStringUntil(' ');
                    kartica(3, "RULE.txt");
                    return Pred;
                }
            }
            else {
                goto DALJE;
            }
        }
        else { // ne poklapa se dan
        DALJE:
            Dan = myFile2.readStringUntil('\n');
            Dan = "";
        }
    }
}

```

```

        goto PONOVI;
    }
}
else return "";
}

// D U M P
void dump() { // funkcija za ispis sadržaja pohranjenog
dokumenta // po default-u je datoteka "PR1.txt" a za
    myFile.seek(0); // bilo koju drugu
    while (myFile.available()) { // potrebno ju je prvo pomoću "text"
otvoriti
        Serial.write(myFile.read());
    }
    kartica(0, imeF);
}

// U P I S I
// funkcija za
void upisi() {
vodjenje dnevnika prisustva
    String dat = String(rtc.getDay()) + "." + String(rtc.getMonth());
    if (kartica(1, imeF)) {
        myFile2.seek(0); //myFile.seek(0);
        int D = traziD(dat);
        if (D > 0) {
            //postoji datum
        }
        else if (D == 0) { // nema datuma
            myFile.print('\n');
UPDAT:
            myFile.print(dat); myFile.println(';');
            Serial.println(F("dat up"));
            Flash(BLUE, 1);
        }
        else if (D < 0) { // nema datuma
            goto UPDAT;
        }
        if (traziS(MBS, D)) { //vec je upisan!
            Serial.println(F("vec si upisan!"));
            Flash(BLUE, 1);
        }
        else { //nisam ga pronasao
            myFile.print(MBS); myFile.print(';');
            Serial.println(F("S up"));
            Flash(GREEN, 1);
        }
    }
    myFile.close(); // Disables SD card chip select pin
    myFile2.close();
    digitalWrite(CS_SD, HIGH);
}

// T R A Z I ( D )
// funkcija koja trazi datum u
unsigned long traziD(String dat) { // funkcija koja trazi datum u
otvorenom myFile-u // trazi od pocetka do prvog ;
    String buff; // zatim preskace cijeli red ispod
    int i = 0; // ukoliko je prazan file vraca -
1 // ukoliko nije pronasao datum
    myFile.seek(0); // vraca 0
    TR_DAT: // ako je pronasao datum vraca
njegovu poziciju poz+1
    if (myFile.position() >= myFile.size()) { //nema ništa -> upiši datum
        if (i == 0) return -1; // odma u startu prazan file -> ne treba \n
        else return 0; // nema u file-u, treba \n
    }
    else { // postoji neki datum
        buff = myFile.readStringUntil(';');

```

```

    i++;
    buff.trim();
    if (buff.equals(dat)) { //pase datum
        return (myFile.position() + 1);
    }
    else { //ne pase, preskoci studente u redu ispod
        myFile.seek(myFile.position() + 2); // preskoci \n nakon datuma
        while (myFile.peek() != '\n' && myFile.position() < myFile.size()) {
            myFile.seek(myFile.position() + 1);
        }
        goto TR_DAT;
    }
}
}

//      T R A Z I ( S )
bool traziS(String mb, int poz) { // funkcija koja trazi MB u
    otvorenom myFile-u od // ulazne pozicije (pozicija
    String buff; // nakon datuma) do kraja
    if (poz <= 0) return false; // ako je tek upisan datum, nema
    S // reda (\n ili dokument)
    myFile.seek(poz + 1); // ukoliko je pronađen MB vraća
    True a inače Flase
    TR_MB:
    if (myFile.position() == myFile.size() || myFile.peek() == '\n' || myFile.peek()
    == '\r') return false;
    buff = myFile.readStringUntil(';');
    buff.trim();
    if (buff.equals(mb)) return true;
    else {
        buff = "";
        goto TR_MB;
    }
}

//      I S P I S I ( D )
void ispisiD(String date) { // funkcija za ispis studenata
    prisutnih na zeljenom predavanju // odrzanom zeljenog datuma
    if (myFile) { // otvara predavanje u myFile a
        unsigned long pozM = traziD(date); // pomocu myFile2
        pretrazivanje stud.txt se vrši // ispisuje studente n;MB;Pre;Ime
        int poz; // završava sa r1 ukoliko su
        String buf; //
        if (pozM > 0) { //ako postoji datum // završava sa e1 ukoliko nije
            ispisani svi studenti //
            Serial.println(date); //
            prepoznalo upisanog studenta
            pozM = pozM + 1;
            poz = 1;
        }
        NEXT:
        myFile.seek(pozM);
        if (myFile.position() == myFile.size() || myFile.peek() == '\n') {
            Serial.print(F("r1"));
            goto END;
        }
        buf = myFile.readStringUntil(';'); pozM = myFile.position();
        buf.trim();
        Serial.print(poz); Serial.print(' ');
        delay(100);
        if (!trazi(buf, -1)) {
            Serial.print(F("r2"));
            goto END;
        }
        digitalWrite(CS_SD, LOW);
        buf = "";
        poz++;
        goto NEXT;
    }
}

```



```

        else {
            Serial.println(F("nema datuma"));
        }
    }
    else Serial.println(F("e1"));
END:
myFile.close();
myFile2.close();
digitalWrite(CS_SD, HIGH);
}

void Foot(int n, String m) {
    Serial.print(F("=AVERAGE(E)")); Serial.print(n); Serial.print(':');
    Serial.print(m); Serial.print(n); Serial.println(F(")*100"));
}

//      T A B L I C A
void Tablica (String pred) { // funkcija za ispisivanje tablice
    prisustva studenata na odabranom // predavanju od početka do zadnjeg
    upisanog datuma
    int d = 1;
    int n = 1;
    int j = 1;
    unsigned long poz;
    digitalWrite(CS_SD, LOW);
    myFile2 = SD.open(studF, FILE_READ);
    myFile = SD.open(pred, FILE_READ);
    Serial.println(F("---")); Serial.print(F("B;MB;Pre;ime;"));
    while (GetDate(d) != "" && GetDate(d) != "\n") {
        Serial.print(GetDate(d)); Serial.print(';');
        d++;
    }
    String E = stupac(d - 1);
    Serial.println('%'); Serial.print(n); Serial.print(' ');
    while (trazi("", n)) {
        Serial.print(' ');
        while (j < d) {
            String dat = GetDate(j);
            dat.trim();
            poz = traziD(dat);
            if (traziS(MBS, poz)) Serial.print('1');
            else Serial.print('0');
            Serial.print(';');
            j++;
            dat = "";
            delay(100);
        }
        n++;
        Foot(n, E);
        Serial.print(n); Serial.print(' ');
        j = 1;
        MBS = "";
        delay(100);
    }
    Serial.print("end");
    myFile2.close(); myFile.close();
    digitalWrite(CS_SD, HIGH);
}

//      D A T U M I
void Datumi(String pred) { // funkcija koja pretražuje sve
    datume u dokumentu predavanja // i zapisuje ih u privremenu
    String buff; // i zapisuje ih u privremenu
    datoteku datum.txt pomocu File-a helpF // ukoliko postoji (od zadnjeg ispisa
    if (kartica(1, pred)) { // ukoliko postoji (od zadnjeg ispisa
        tablice), potrebno // ju je pobrisati i ispisata
        if (SD.exists(datum)) SD.remove(datum); // ju je pobrisati i ispisata
    }
    aktualne datume
}

```

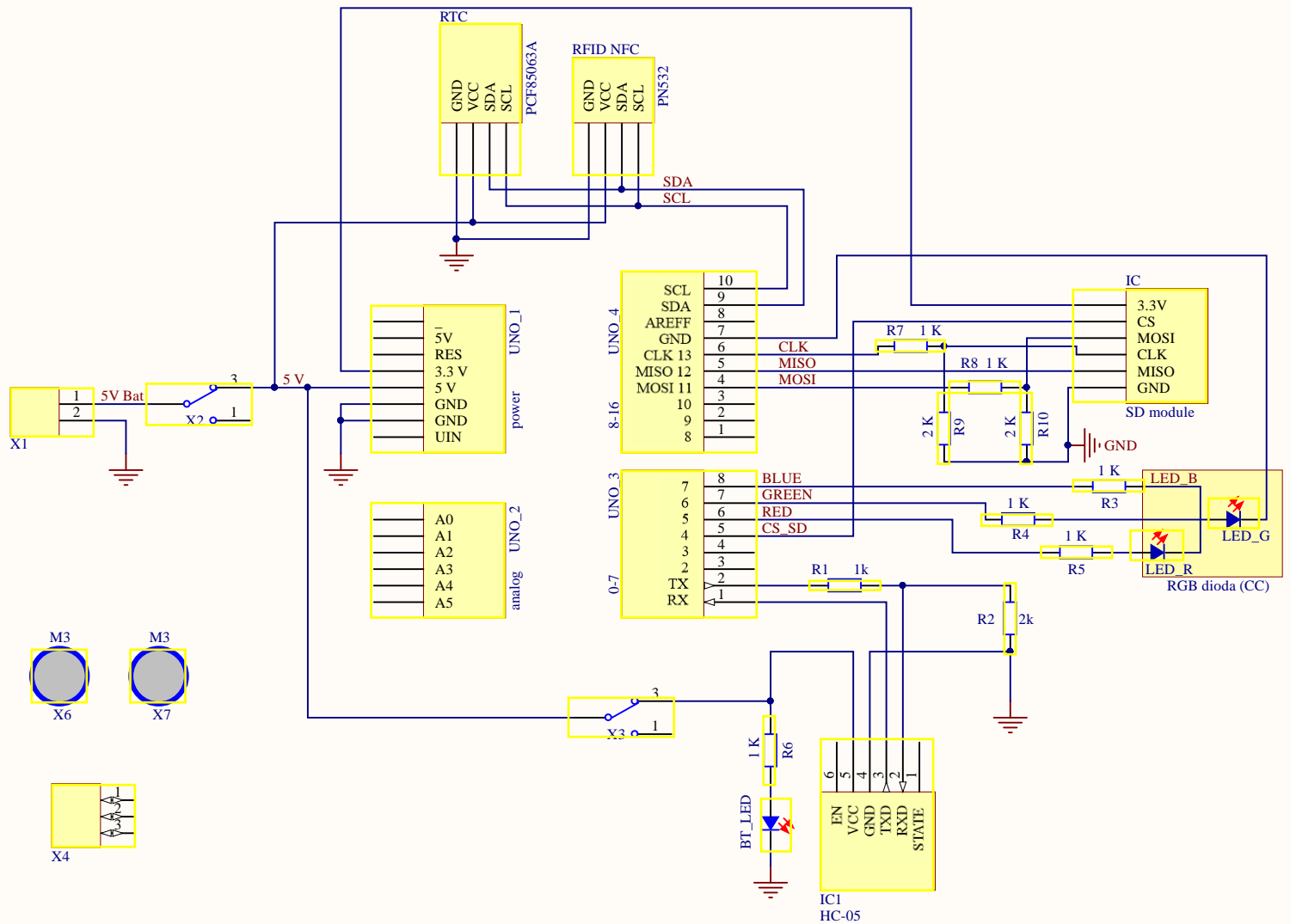
```

    helpF = SD.open(datum, FILE_WRITE);
    if (helpF) {
PONOVO:
        if (myFile.position() == myFile.size()) {
            helpF.close();
        }
        else {
            buff = myFile.readStringUntil('\n'); // procitaj datum
            helpF.print(buff);
            while (myFile.peek() != '\n' && myFile.position() != myFile.size()) { //
preskoci studente
                myFile.seek(myFile.position() + 1);
            }
            if (myFile.position() != myFile.size()) myFile.seek(myFile.position() + 1);
            goto PONOVO;
        }
    }
}

// GET DATE
String GetDate(int A) { // funkcija koja otvara datum.txt i
    cita A.ti datum po redu // vraca datum, ukoliko nema vraca
    String out; // mora biti ukljuceno citanje kartice
    prazno ("")
    int i = 0; // mora biti ukljuceno citanje kartice
    jer ova funkcija to ne pali!
    helpF = SD.open(datum, FILE_READ); // --> digitalWrite(CS_SD, LOW);
    if (helpF) {
        while (helpF.position() != helpF.size()) {
            out = helpF.readStringUntil(';');
            out.trim();
            i++;
            if (i == A) {
                helpF.close();
                return out;
            }
            out = "";
        }
        helpF.close();
        return "";
    }
}

void loop() { // L O O P
    do { // čekaj karticu
        if (Serial.available() > 0) {
            delay(20);
            citajP();
        }
        success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength,
TimeOut);
    }
    while (!success);
    String uidString = "";
    for (uint8_t i = 0; i < 4; i++) {
        uidString += uid[i];
        if (i != 3) uidString += ' ';
    }
    do // čekaj uklanjanje kartice
        success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength,
TimeOut);
    while (success);
    test("a");
    test(uidString);
    delay(250);
}

```



Naziv Shema Citaca	Fakultet strojarstva i brodogradnje	<b>MIR</b>
Student Patrik Putanec 0035 219628	Sveučilište u Zagrebu	
Datum: 24.2.2022. Vrijeme: 3:14:43	Ivana Lučića 5	
File: C:\Obrazovanje\ZAVRSNI\Altium\CitacF1\ShemaCitaca.SchDoc	10 000 Zagreb	

1

2

3

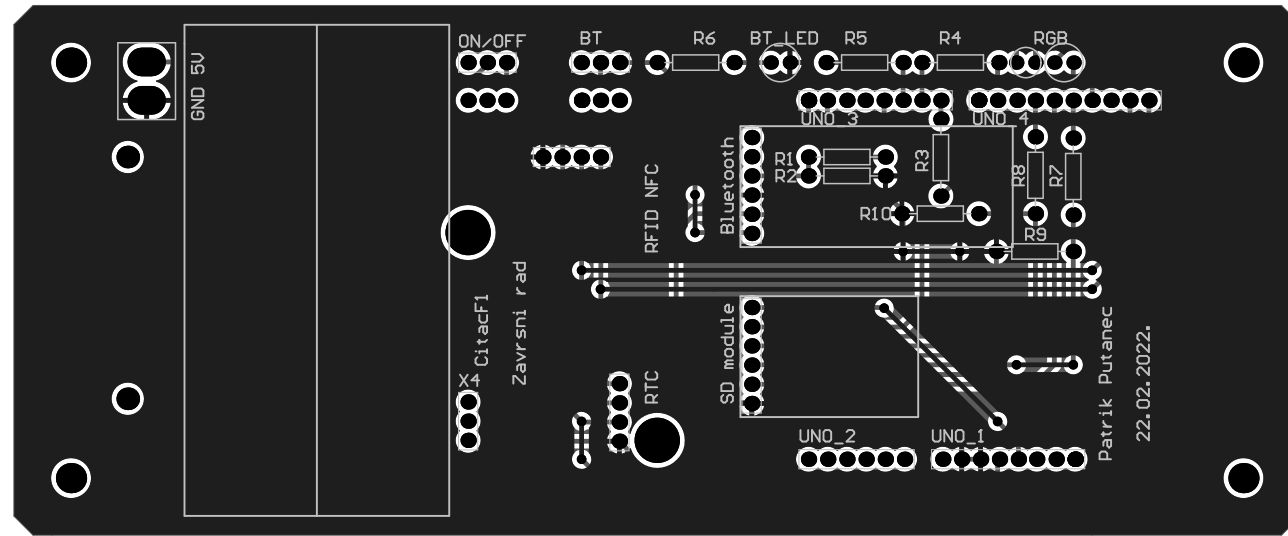
4

A

B

C

D



# MiR

Fakultet strojarstva i brodogradnje  
Sveučilište u Zagrebu  
Ivana Lucica 5  
10 000 Zagreb

STUDENT: Patrik Putanec

PROJEKT: Završni rad

DATUM: 24.2.2022.

FILE: C:\Obrazovanje\ZAVRSNI\Altium\CitacF1\CitacPCB\_v2.PcbDoc

NAZIV:  
CitacF1

KOMENTAR:

1

2

3

4

A

B

C

D