



UNIVERZITET U BIHAĆU  
TEHNIČKI FAKULTET  
ODSJEK ZA ELEKTROTEHNIKU

---

# **Sistem za onemogućavanje upravljanja službenim vozilom, od strane osoba u alkoholisanom stanju**

---

PROJEKTN A DOKUMENTACIJA IZ PREDMETA  
- SISTEMI U REALNOM VREMENU -

**Student:**  
**Ivan Ivušić**

**Profesor: Vanr. prof. dr Edin Mujčić**  
**Asistent: mr. Una Drakulić**

Bihać, sept 2020.

# Sadržaj

<b>Popis slika</b>	<b>ii</b>
<b>1 Uvod</b>	<b>1</b>
<b>2 Dizajn i implementacija</b>	<b>2</b>
2.1 Maketa sistema . . . . .	2
2.2 Korištene komponente . . . . .	7
2.2.1 Arduino Uno ploča . . . . .	7
2.2.2 GPS modul . . . . .	8
2.2.3 GSM modul . . . . .	8
2.2.4 Bluetooth modul . . . . .	9
2.2.5 Zujalica . . . . .	9
2.2.6 Alkoholni senzor . . . . .	10
2.2.7 RFID sistem . . . . .	10
2.2.8 Releji . . . . .	11
2.2.9 Eksperimentalna pločica . . . . .	11
<b>3 Programiranje sistema</b>	<b>12</b>
3.1 Arduino IDE . . . . .	12
3.2 MIT app inventor . . . . .	13
3.3 Arduino kod . . . . .	14
3.4 Blokovski kod . . . . .	20
3.5 Komunikacija . . . . .	21
<b>4 Eksperimentalna analiza rada sistema</b>	<b>23</b>
<b>5 Zaključak</b>	<b>26</b>
<b>Prilozi</b>	<b>27</b>
<b>A Kodni program projekta</b>	<b>28</b>
A.1 Arduino kod . . . . .	28
<b>Literatura</b>	<b>34</b>

# Popis slika

2.1	Prikaz makete sistema . . . . .	2
2.2	Prikaz unutrašnjeg dijela makete . . . . .	3
2.3	Prikaz prve cjeline sklopa u Fritzing softveru. . . . .	4
2.4	Prikaz prve cjeline električne sheme sklopa. . . . .	5
2.5	Prikaz dizajna aplikacije. . . . .	5
2.6	Prikaz druge cjeline sklopa u Fritzing softveru. . . . .	6
2.7	Prikaz električne sheme druge cjeline sklopa. . . . .	6
2.8	Arduino Uno pločica. . . . .	7
2.9	GY-NEO 6M Gps modul. . . . .	8
2.10	SIM900 GPS/GPRS modul. . . . .	8
2.11	Prikaz bluetooth modula. . . . .	9
2.12	Prikaz zujalice. . . . .	9
2.13	Prikaz alkoholnog MQ3 senzora. . . . .	10
2.14	Prikaz RFID modula. . . . .	10
2.15	Prikaz releja. . . . .	11
2.16	Prikaz eksperimentalne pločice. . . . .	11
3.1	Prikaz Arduino IDE-a. . . . .	12
3.2	Prikaz dizajn dijela Inventora. . . . .	13
3.3	Prikaz blok dijela Inventora. . . . .	14
3.4	Prikaz prve grupe blokovske sheme mobilne aplikacije. . . . .	21
3.5	Prikaz druge grupe blokovske sheme mobilne aplikacije. . . . .	21
3.6	Prikaz treće grupe blokovske sheme mobilne aplikacije. . . . .	21
3.7	Prikaz UART komunikacije. . . . .	22
3.8	Prikaz SPI komunikacije. . . . .	22
4.1	Prikaz sistema u stanju blokiranja. . . . .	23
4.2	Prikaz sistema u dopuštenom stanju. . . . .	24
4.3	Prikaz baze podataka sa GPS podacima. . . . .	24
4.4	Prikaz prozora za registraciju. . . . .	25
4.5	Prikaz GPS podataka na stranici. . . . .	25

# Poglavlje 1

## Uvod

Kroz ovaj seminarski rad će se prikazati postupak kreiranja sistema za onemogućavanje upravljanja službenim vozilom, od strane osoba koje su u bilo kakvoj vrsti alkoholisanog stanja. Svakom ekspiracijom subjekta dolazi do izbacivanja alkoholnih para, ukoliko je osoba konzumirala bilo kakvu vrstu alkoholnog pića. Broj nesreća izazvanih vožnjom u alkoholisanom stanju svakodnevno uzimaju velik broj života, međutim do današnjeg dana automobili nisu opremljeni mehanizmima zabrane vožnje za nemarne vozače.

Senzor koji će koristiti za kupljenje podataka o parama iz svoje okoline je MQ-3 senzor. On nije kreiran za korištenje u ovu svrhu, ali može poslužiti kao dovoljno validan uređaj za detekciju u ovoj prilici. Kako je ovaj sistem namjenjen za službena vozila, svaki korisnik će morati izvršiti dodatni proces autorizacije, korištenjem RFID modula. Svi radnici dobijaju po jednu identifikacionu karticu koju stavljaju u posebno namjenjenu poziciju, davajući sistemu svoje podatke.

Kako bi se znalo kretanje službenih vozila, sistem omogućuje kupljenje trenutne GPS lokacije automobila, i njeno slanje na internet server. Ova funkcija služi kao dodatni izvor informacija u slučaju krađe automobila.

Pretpostavka je da će krajnja funkcionalnost biti ograničena, pogotovo u pogledu tačnog identificiranja stanja korisnika. Korišteni senzori nisu namjenjeni za korištenje u profesionalne svrhe, kao ni za ove specificirane uvjete, ali mogu poslužiti u demonstrativne svrhe.

## Poglavlje 2

# Dizajn i implementacija

U ovoj cjelini će se reći par stvari o samom dizajnu sistema, korištenim komponentama, kao i pripadnom mikrokontroleru koji usklađuje rad svih cjelina.

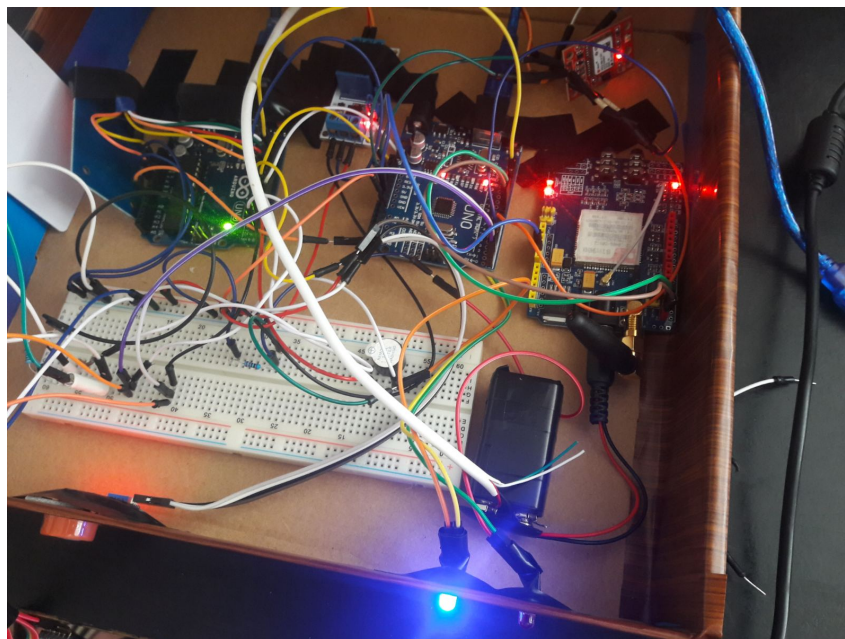
### 2.1 Maketa sistema

Sistem je osmišljen kao crna kutija (engl. Black Box), sa dva glavna izlaza, koja služe za napajanje komponenti, te sa dva priključka koja omogućuju vezu akumulatora sa motorom auta. Samo djelovanje na priključak, bilo ono omogućavajuće u smislu protoka struje ili ne, čini glavni rezultatnu akciju svih podržanih funkcija. Pošto je riječ o studentskom radu, kućište sistema je izgrađeno korištenjem kartona, na kojeg je postavljena prikladna prevlaka. Prikaz finalnog proizvoda je vidljiv na slici 2.1.



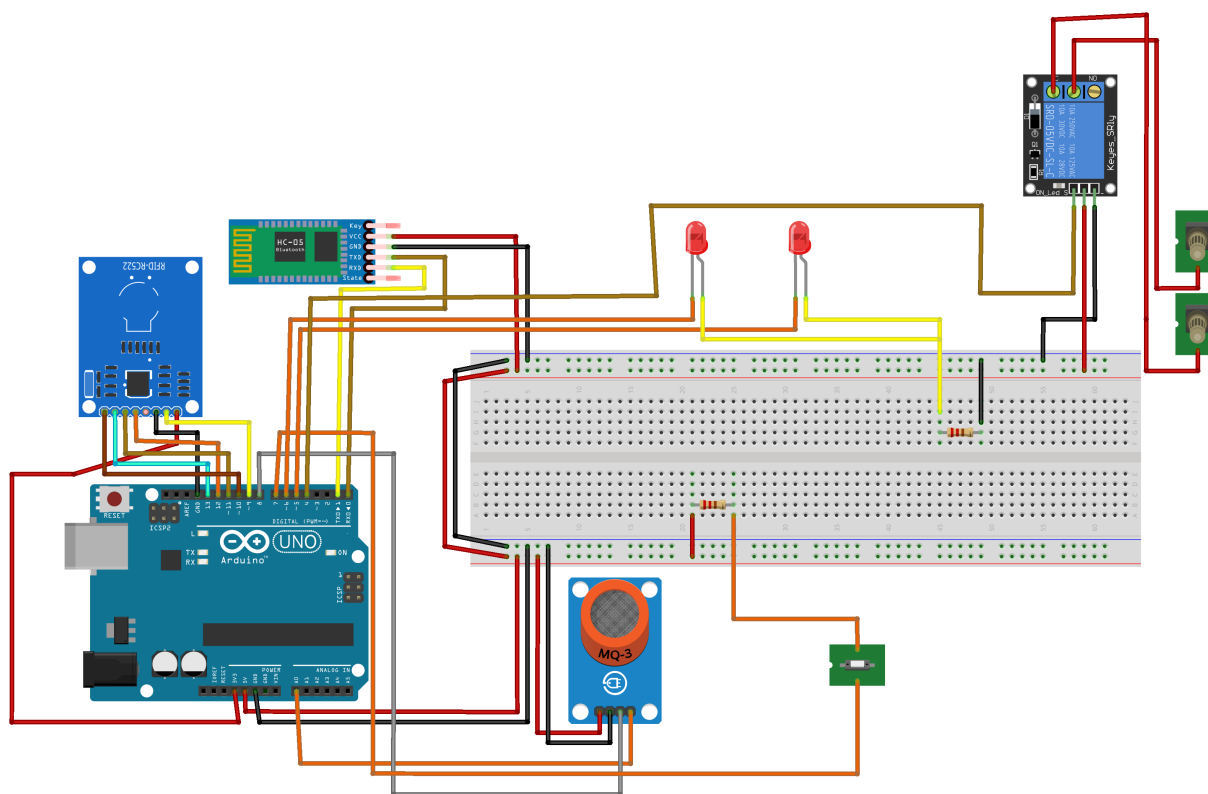
**Slika 2.1:** Prikaz makete sistema

Prikaz spojeva sistema na prvu djeluje veoma komplicirano, kao što je moguće vidjeti na slici 2.2, ali shema je prilično jednostavna. Svaki element sistema je pričvršćen za podlogu korištenjem izolirne trake, te je tako sistem otporan na razumnu količinu vibracije, i drmanja.



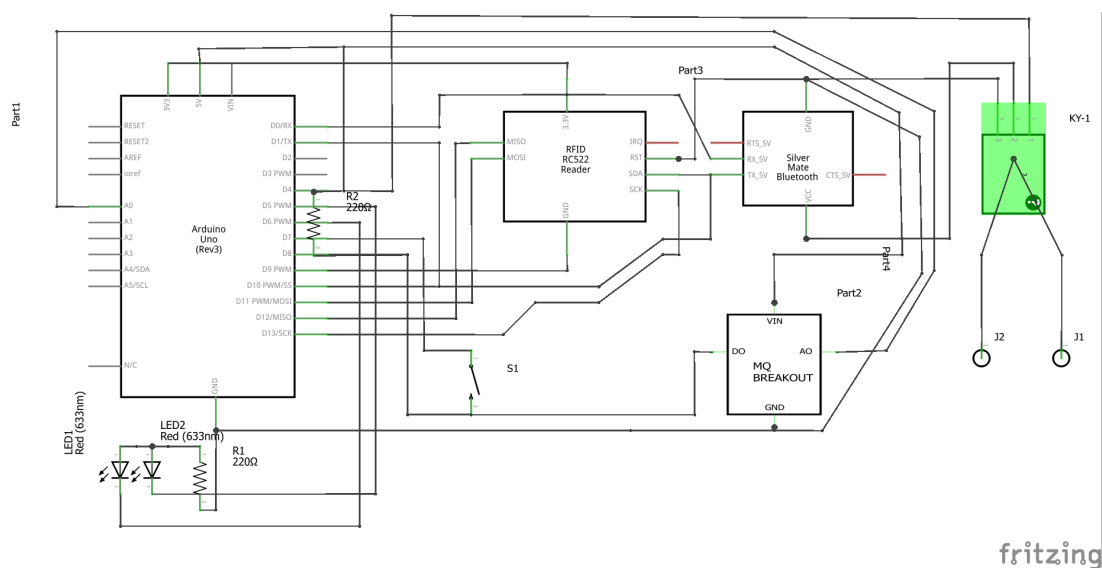
**Slika 2.2:** Prikaz unutrašnjeg dijela makete

Sistem je napajan korištenjem USB interfejsa radi stabilnosti ulaznog napona, čiji priključak nudi svaki moderni auto. Kratkim modifikacijom sistema je moguće ostvariti direktno napajanje putem akumulatora, ali radi lakšeg demonstrativnog procesa je taj način odbačen. Shema sklopa, iako je prilično jednostavna, sadrži dosta komponenti. Hardver se sastoji od dvije Arduino ploče, koje komuniciraju serijskom vezom. Radi lakšeg prikaza sklop je podijeljen u dvije cjeline, koje odgovaraju modulima sa Arduino Uno pločama na koje su priključene. Sve sheme su kreirane softverom "Fritzing".



Slika 2.3: Prikaz prve cjeline sklopa u Fritzing softveru.

Kao što je vidljivo na slici 2.3., na Arduino Uno ploču je povezan RFID čitač, koji vrši funkciju provjere valjanosti identifikacije osobe, koja je postavila svoju oznaku unutar zone mjerenja. Analogni oblik izlaza, sa MQ-3 senzora, je priključen na prvi analogni ulaz Uno pločice. Ista vrijednost se koristi za provjera količine alkoholne pare u zraku. Na nulti i prvi digitalni ulaz je povezan bluetooth modul, tj. na pin za primanje i slanje serijske informacije. Isti šalje pametnom uređaju informaciju kad da posalje GPS podatke serveru. Nakon odobrenja paljenja auta, od strane sistema, šalje se visoko stanje relej, koji omogućuje nesmetanu vezu akumulatora sa motorom. Kako bi se dalo doznajanja sistemu da se auto želi pokrenuti, pritiska se dugme za paljenje. Prikaz električne sheme je vidljiv na slici 2.4.



**Slika 2.4:** Prikaz prve cjeline električne sheme sklopa.

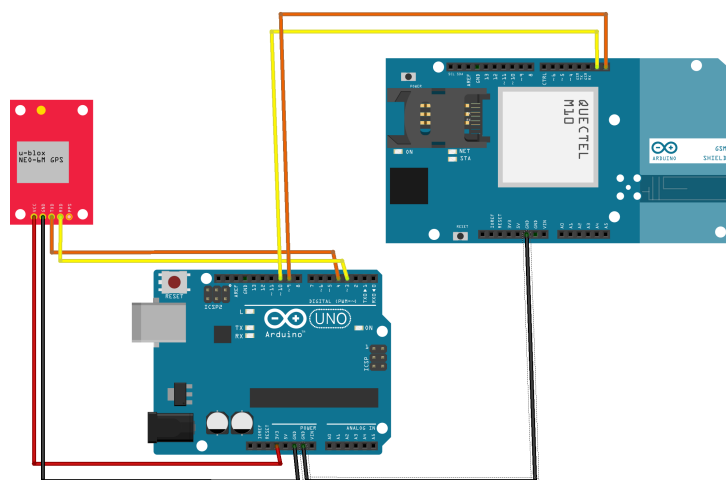
Kako "Fritzing" ne nudi prikaz određenih komponenti, tj. bluetooth modula i alkoholnog senzora, iste su preuzete sa oficijelne "Fritzing" stranice. Korištenjem aplikacije, prikazane na slici 2.5., korisnik unosi UID vozila, i time omogućava slanje podataka o lokaciji na predefini-sani server. Poticaj za slanje dolazi sa bluetooth modula, u obliku karaktera "1".



**Slika 2.5:** Prikaz dizajna aplikacije.

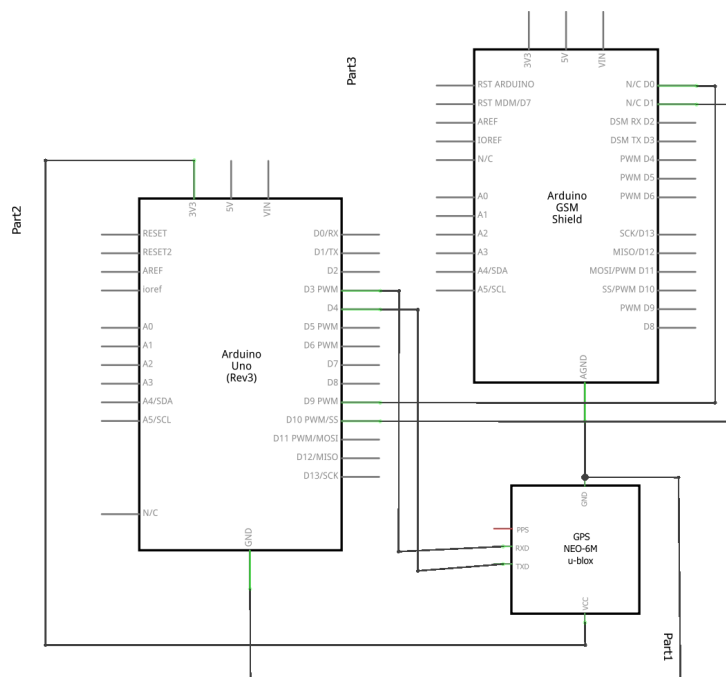
Druga cjelina se sastoji od GSM i GPS modula, priključenih na Arduino ploču, kako je već naznačeno. Određenom modifikacijom moguće je sistem svesti na jednu cjelinu, ali radi lakšeg procesa implementacije su funkcije razdvojene. Prikaz druge cjeline je vidljiv na slici 2.6.





**Slika 2.6:** Prikaz druge cjeline sklopa u Fritzing softveru.

GPS, i GSM, moduli su priključeni na digitalne ulaze, gdje serijskom komunikacijom se vrši prijenos podataka. GPS modul služi za primanje SMS poruka, sa tačno određenog broja, te u ovisnosti od sadržaja poruke vrše određene funkcije. Ukoliko je sadržaj poruke "Ugasi", tj. "Upali", vrši se promjena stanja veze akumulatora i motora. Prikaz električne sheme je vidljivi na slici 2.7.



**Slika 2.7:** Prikaz električne sheme druge cjeline sklopa.

Kako "Fritzing" ne nudi prikaz određenih komponenti, tj. GPS i GSM modula, iste su preuzete sa oficijelne "Fritzing" stranice.

## 2.2 Korištene komponente

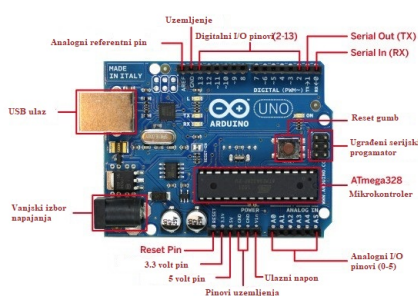
Korištene komponente, njihova funkcija, kao i količina su opisani u ovoj sekciji. Za izradu ovog projekta je potrebno imati:

- Arduino Uno - 2x
- Gps modul - 1x
- Gsm modul - 1x
- Bluetooth modul - 1x
- Dugme - 1x
- Zujalica - 1x
- Alkoholni senzor - 1x
- 1000 i 220  $\Omega$  Otpornik - 1x
- Releji - 1x
- RFID sistem - 1x
- Eksperimentalna pločica - 1x
- Kablovi

Tip, i model svake kompliciranije komponente je dan u nastavku.

### 2.2.1 Arduino Uno ploča

Arduino Uno je mikrokontrolerska ploča zasnovana na ATmega328 mikrokontroleru. Prikaz iste je vidljiv na slici 2.8.

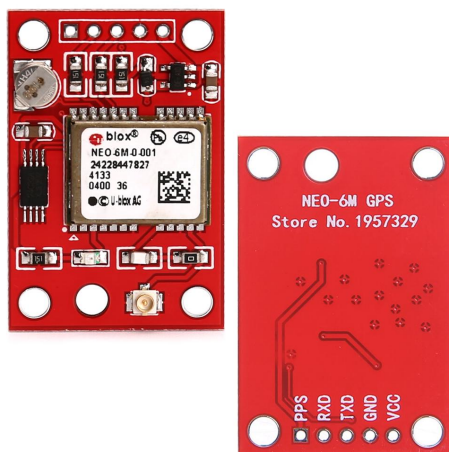


**Slika 2.8:** Arduino Uno pločica.

Ima 14 digitalnih ulazno / izlaznih pinova, od kojih se 6 mogu koristiti kao PWM izlazi, 6 kao analogni ulazi. Takt mu obezbeđuje keramički oscilator od 16 MHz, te posjeduje USB vezu, utičnicu za napajanje, ICSP zaglavlje i dugme za resetiranje. Uno se razlikuje od svih prethodnih ploča po tome što ne koristi FTDI USB-to-serial driver čip. Umjesto toga, sadrži Atmega16U2 čip programiran kao USB-to-serial pretvarač[1].

### 2.2.2 GPS modul

GPS moduli sadrže male procesore i antene koji direktno primaju podatke poslane od satelita putem namjenskih RF frekvencija. Odatle se prima vremensku oznaku sa svakog vidljivog satelita, zajedno s ostalim dijelovima podataka. Ako antena modula može uočiti 4 ili više satelita, u stanju je precizno izračunati svoj položaj i vrijeme. Prikaz modula je vidljiv na slici 2.9.



**Slika 2.9:** GY-NEO 6M Gps modul.

Pomenuti modul radi pri napajanju od 3.3 V, te ima osnovnu brzinu komunikacije od 9600 bita po sekundi.

### 2.2.3 GSM modul

GSM modul ili GPRS modul je čip ili sklop koji će se koristiti za uspostavljanje komunikacije između mobilnog uređaja ili računarske mašine i GSM ili GPRS sistema. GSM (Globalni sistem za mobilne komunikacije), standard je koji je razvio Evropski institut za telekomunikacijske standarde (ETSI). Prikaz štita je vidljiv na slici 2.10.



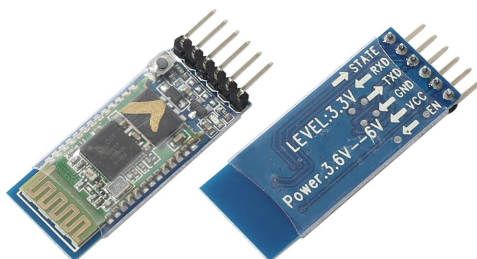
**Slika 2.10:** SIM900 GPS/GPRS modul.

Kako je bilo problema u nabavki modula, objedinjena funkcija je realizirana korištenjem

androidno pogonjenog mobitela, uz Mit App Inventor aplikaciju, kao i korištenjem prikazanog modula.[2].

### 2.2.4 Bluetooth modul

Veza mobilnog računala i vozila će se ostvariti korištenjem bežične, serijske veze. Korišteni bluetooth modul je HC-05 tipa, te je prikazan na slici 2.11.



**Slika 2.11:** Prikaz bluetooth modula.

Osnovna baud brzina modula je 9600 bitova po sekundi, te ima pokrivenost od 10 metara. Ulazni napon se kreće u vrijednosti od 3.3 do 6V DC, dok je operacioni napon jednak 3.3V DC [3].

### 2.2.5 Zujalica

Zujalica je elektromehanički uređaj koji radi na principu piezoelektričnog efekta, i korišten je za stvaranje tonova u aplikacijama gdje su potrebne jednostavne melodije. Prikaz je vidljiv na slici 2.12.

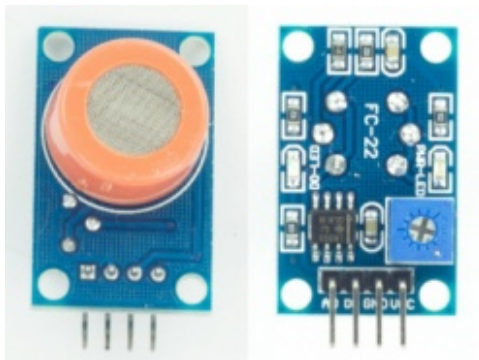


**Slika 2.12:** Prikaz zujalice.

Uređaj se napaja naponom od 5V, pri čemu crpi struju od 15mA. Daje veoma specifičan ton [4].

### 2.2.6 Alkoholni senzor

Alkoholni senzor je uređaj koji služi za mjerenje količine alkoholne pare u okolnom zraku. Korišteni tip senzora je MQ3 je senzor, koji reagira na etanol. Daje digitalne i analogne izlaze očitavanja alkohola u zraku. Prikaz senzora je vidljiv na slici 2.13.



**Slika 2.13:** Prikaz alkoholnog MQ3 senzora.

Spajanjem na 5V se pali grijač koji povlači maksimalno 150mA. Drugi dio ovog senzora je promjenjivi otpornik. Otpor unutar senzora se mijenja s obzirom na količinu alkohola u zraku, na način da više alkohola znači manji otpor. Na kraju se izmjeri napon na analognom pinu, te veće vrijednosti očitavanja označavati veću količinu alkohola u zraku[5].

### 2.2.7 RFID sistem

RFID modul vrši identifikaciju pomoću radio-frekvencije. Isti koristi elektromagnetska polja za prijenos podataka na kratke udaljenosti. Informacije za identifikaciju nosi iskaznica, u obliku kartice, koja mora biti korištena od strane jednog, i samo jednog, korisnika. Prikaz RFID čitača i kartice je vidljiv na slici 2.14.



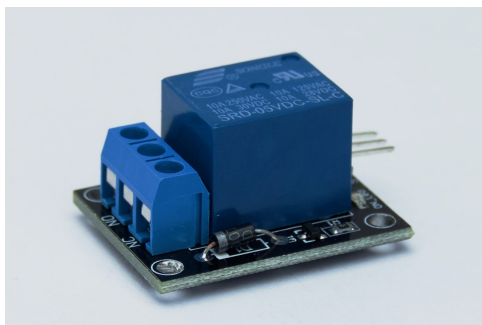
**Slika 2.14:** Prikaz RFID modula.

RFID čitač se sastoji od radio frekvencijskog modula, upravljačke jedinice i antenske zavojnice koja generira visokofrekventno elektromagnetno polje. S druge strane, oznaka je obično pasivna komponenta koja se sastoji od samo antene i elektroničkog mikročipa, pa kad se približi

elektromagnetskom polju primopredajnika, uslijed indukcije, u njenoj zavojnici antene stvara se napon i to napon služi kao snaga za mikročip. Napajanje je vrijednosti od 3.3V DC, uz frekvenciju rada od 13.56MHz[6].

### 2.2.8 Relej

Relej je prekidač na električni pogon koji omogućuje prolazak, odnosno blokira prolazak, struje. Kontroliran je niskim naponima, poput 5V koje pružaju Arduino pinovi. Upravljanje relejnim modulom s Arduinoom jednostavno je kao i upravljanje bilo kojim drugim izlazom.

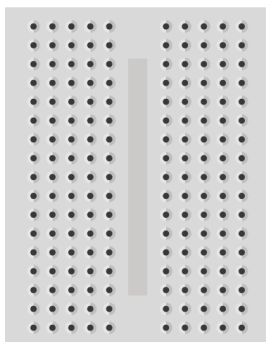


**Slika 2.15:** Prikaz releja.

Ovaj modul, slika 2.15., treba napajati s 5 V, što je prikladno za upotrebu sa Arduinoom. Postoje i drugi relejni moduli koji se napajaju pomoću 3.3V, što je idealno za ESP32, ESP8266 i druge mikrokontrolere [7].

### 2.2.9 Eksperimentalna pločica

Eksperimentalna pločica predstavlja osnovnu podlogu za kreiranje elektronskih prototipa. Budući da ploča ne zahtijeva lemljenje, može se ponovo upotrijebiti. To ga olakšava za upotrebu privremenih prototipa i eksperimentiranje sa dizajnom kola. Prikaz ploče je vidljiv na slici 2.16.



**Slika 2.16:** Prikaz eksperimentalne pločice.

U ovoj verziji pločice je svaki red utora na zasebnim stranama povezan.

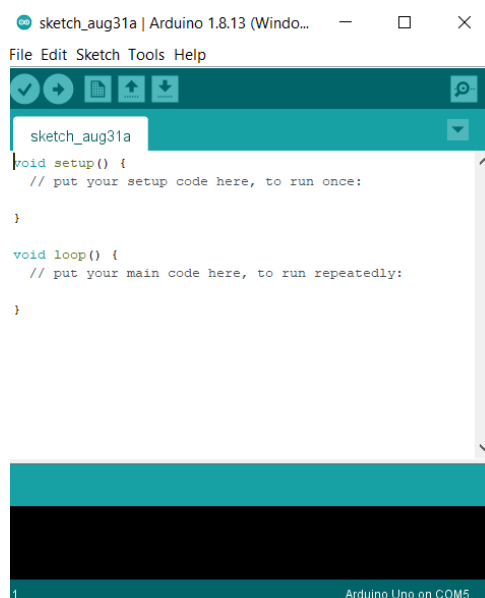
## Poglavlje 3

# Programiranje sistema

U ovoj cjelini će se objasniti korišteni program, koji reguliše rad sistema. Detaljnije objasniti kod, kao i logika iza njegovog rada. Pored ovog, će se prvo nešto reći o korištenim razvojnim okruženjima.

### 3.1 Arduino IDE

Arduino integrirano razvojno okruženje predstavlja aplikaciju koja se koristi za pisanje i učitavanje programa na kompatibilne Arduino ploče. Omogućeno je korištenje iste na više platformi, kao što su Windows, macOS, Linux, te je napisana u funkcijama iz C i C ++. Izvorni kôd za IDE objavljen je pod generalnom, javnom licencom, te podržava jezike C i C ++ koristeći posebna pravila strukturiranja koda.



**Slika 3.1:** Prikaz Arduino IDE-a.

Svaka skica, kao prikazana na slici 3.1., uvijek ima dva glavna dijela, void setup i void loop rutine. Prva rutina započinje sa radom pri paljenju Arduino ploče. Ova se funkcija izvršava

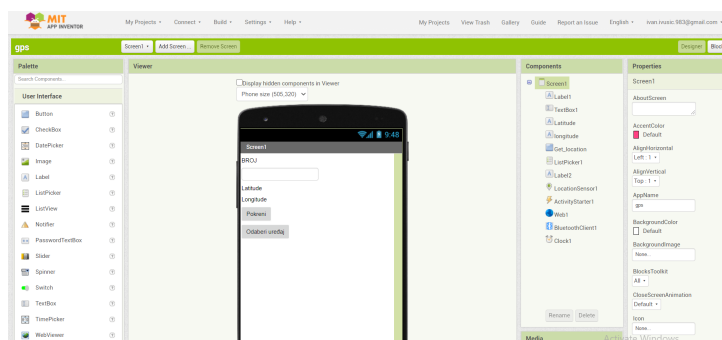
samo jednom tokom cijelog funkcioniranja programa. Funkcija podešavanja sadrži inicijalizaciju svakog pina koji namjerava koristiti u projektu, za ulaz ili izlaz, kao i inicijalizaciju serijskog porta. Naredna rutina se sastoji od onog dijela koda koji se treba kontinuirano izvršavati, za razliku od dijela koda napisanog u prvoj rutini.

Arduino IDE isporučuje softversku biblioteku iz projekta "Wiring", koji pruža brojne uobičajene procedure unosa i izlaza. Arduino IDE koristi program "avrdude" za pretvaranje izvršnog koda u tekstualnu datoteku u heksadecimalnom kodiranju koja se učitava na ploču Arduino pomoću programa za učitavanje u firmwaeu ploče. Prema zadanim postavkama, avrdude se koristi kao alat za prijenos za prebacivanje korisničkog koda na službene Arduino ploče. S porastom popularnosti Arduina kao softverske platforme, drugi dobavljači su počeli primjenjivati prilagođene kompajlere i alate otvorenog koda koji mogu graditi i slati skice na druge mikrokontrolere koji nisu podržani od strane Arduinove službene linije mikrokontrolera.

## 3.2 MIT app inventor

MIT App Inventor je razvojno okruženje koje omogućava lagano kreiranje aplikativnog softvera za dva operativna sistema, Android i iOS. Isti je prvobitno pružio Google, ali sada ga održava Massachusetts Institute of Technology, u obliku internet aplikacije. Aplikacije se kreiraju na vizualni način, povlačenjem i spuštanjem objekata, što, unatoč smanjenju potpune upravljivosti nad kodom funkcija, povećava brzinu proizvodnje aplikacije

Koristi grafičko korisničko sučelje vrlo slično programskim jezicima Scratch i StarLogo, te omogućava korisnicima povlačenje i ispuštanje vizualnih objekata kako bi stvorili aplikaciju koja može raditi na android uređajima, dok App-Inventor Companion koji radi na iOS pokrenutim uređajima još uvijek je u beta fazi. MIT App Inventor uključuje dva glavna prikaza: prikaz dizajna i prikaz blokova. Prikaz dizajna je vidljiv na slici. Sučeljem se interaktuje povlačenjem i ispuštanjem elemenata korisničkog sučelja aplikacije, te pozicioniranjem istih na željenu lokaciju. Prikaz je vidljiv na slici 3.2.



Slika 3.2: Prikaz dizajn dijela Inventora.

Izumitelji aplikacija vuku komponente iz palete, pozicionirane krajnje lijevo, u preglednik, pozicioniran u sredini, da bi ih dodali u aplikaciju.

Uređivač blokova, prikazan na slici, je okruženje u kojem izumitelji aplikacija mogu vizualno postaviti logiku svojih aplikacija pomoću blokova kodiranih bojom koji se spajaju poput





Dalje su definirane sve zastavice, kao i tajmeri, koji su korišteni u uslovima grananja. Oni omogućavaju precizan rad sistema.

**Program 3.3:** Inicijalizacija kontrolnih varijabli

```
1 // kontrolne zastavice i tajmeri
2 int flagKontrolni = 0; // za omogucavanje nesmetanog rada propusne
  veze akumulator-motor
3 int flagAlkohol = 0; // zastavica za
4 int brojac = 0; // koristen za mjerenja broja zvučnih
  signalizacija
5 float timer1 = 0; // koristen za slanje GPS poticaja
6 int GSMflag = 1; // upravljanje propusnim stanjem putem SMS poruke
```

Potom su inicijalizirane varijable za spremanje UID skenirane kartice, te za spremanje sadržaja SMS poruke, kojom se, u ovisnosti od sadržaja, mijenja stanje propusnosti veze akumulatora sa motorom. Zadnja prikazana varijabla služi za definiranje maksimalnog dozvoljenog nivoa izlaza sa alkoholnog senzora, prije nego što sistem odluči pokrenuti zvučnu signalizaciju, te isključenje vozila.

**Program 3.4:** Inicijalizacija tekstualnih varijabli

```
1 String content = ""; //Varijabla za spremanje identifikacione rijeci
2 String SMSporuka="";
3 //varijabla za spremanje nivoa sa alkoholnog senzora
4 int limit;
```

Naredni kod prikazuje "setup()" funkciju, koja služi za inicijalizaciju serijske veze, bilo softverske ili hardverske, te naredbu za pokretanje rada RFID čitača. Definiraju se tipovi priključaka, da li su ulazni ili izlazni, zatim se prema vrijednost trenutnog relativnog vremena, koje će biti korišteno kao uvjet za slanje GPS podataka. Prvobitno stanje propusnosti akumulatora i motora je blokirano, što je prikazano signalnim svjetlima, preko funkcije "flags()".

**Program 3.5:** Postavna funkcija

```
1 void setup()
2 {
3   Serial.begin(9600); // pokretanje serijske komunikacije
4   SPI.begin(); // pokretanje softverske serijske veze
5   mfrc522.PCD_Init(); // pokretanje RFID citaca
6   //definiranje ulaza/izlaza
7   pinMode(DOUTpin, INPUT);
8   pinMode(pass, OUTPUT);
9   pinMode(block, OUTPUT);
10  pinMode(relay, OUTPUT);
11  pinMode(buzzer, OUTPUT);
12  pinMode(kljuc, INPUT);
13  flags(0, 1, 0); //inicijalizacija signalnih svjetala
14  timer1 = millis();
15 }
```

Trenutni blok koda služi za kontrolu propusnosti akumulatora ka motoru, u ovisnosti od sadržaja teksta primljenog serijskom vezom sa druge Arduino pločice. Ukoliko tekst sadrži riječ "iskljuci", veza sa motorom se blokira, i vozilo postaje neupravljivo, do trenutka stizanja teksta sadržaja "ukljuci".

**Program 3.6:** Blok za kontrolu SMS porukom

```
1 //blok za kontrolu stanja putem SMS poruke
```

```

2   if (Serial.available()>0){
3       SMSporuka=Serial.readString();
4       if (SMSporuka.indexOf("iskljuci") > 0){
5           GSMflag = 0;
6           SMSporuka="";
7
8       }
9       if (SMSporuka.indexOf("ukljuci") > 0){
10          GSMflag = 1;
11          SMSporuka="";
12      }
13  }

```

Blok koda prikazuje slanje karaktera "1", pod uslovom da je od prošlog slanja prošlo minimalno deset sekundi. Karakter signalizira android aplikaciji da je vrijeme za prikupljanje GPS podataka trenutne lokacije, i slanje iste korištenjem GSM mreže. Drugi blok pokreće RFID čitač na skeniranje kartice, ukoliko je nova detektovana. Čitanje se je implemenitirano funkcijom "citanjeKartice()".

## Program 3.7: Blokovi za kontrolu aplikacije i RFID citaca

```

1   // Slanje impulsa uredjaju za kupljenje GPS podataka, i slanje
    istih
2   if (millis() - timer1 > 10000) {
3       Serial.println("1");
4       timer1 = millis();
5   }
6
7   //pozivanje citaca, ukoliko je nova kartica prisutna
8   if (mfrc522.PICC_IsNewCardPresent()) {
9       citanjeKartice()

```

Naredni blok sadrži glavninu logike iza ovog sistema. Sistem omogućava propusnost veze akumulatora sa motorom ukoliko je zadovoljen uslov iz bloka. Korisnik mora imati karticu tačno specifikiranog UID-a, te mora iskazati želju za paljenjem sistema preko dugmeta sa dva stanja. Pri svakoj iteraciji se također provjerava zastavica koja daje uvid u prekoračenju dozvoljenog stanja nivoa alkoholnih para, te zastavica za GSM upravljanje vozilom. Zatim se čisti memorija RFID čitača, i postavlja zastavica za nesmetan ulazak u trenutni blok, pod uvjetom da je dugme za paljenje u niskom stanju. Tokom svake iteracije se očitava izlaz alkoholnog senzora, te ukoliko se nivo premaši pokreće se zvučna signalizacija putem zujalice, i onemogućuje veza sa motorom. Stanja propusnosti, koja su rezultat ove logike, su prikazana korisniku signalnim svjetlima, koja se realiziraju "flags()" funkcijom.

## Program 3.8: Glavni izvršni blok

```

1
2   if (((content=="_33_87_3B_39" && digitalRead(kljuc) == LOW) || (
    digitalRead(kljuc) == LOW && flagKontrolni == 1)) &&
    flagAlkohol == 0 && GSMflag==1){
3       flagKontrolni = 1;
4       for (uint8_t i = 0; i < mfrc522.uid.size; i++) { // brisanje
        buffera RFID citaca
5       mfrc522.uid.uidByte[i] = 0;
6       content="";
7   }
8   value = analogRead(AOUTpin); //dobijanje izlaza mq3 senzora
9   delay(100);

```

```

10 // ukoliko se iskaze zelja gasenja, isto se gasi
11 if (digitalRead(kljuc) == HIGH) {
12     flagKontrolni = 0;
13     flags(0, 1, 0);
14 }
15 // ukoli je kolicina alkoholne pare veca od nivoaa, pokrene se
    alarm zujalicom, i nakon odredjenog vremena se auto gasi
16 if (value > 857) {
17     while(true){
18         //Serial.print(brojac);
19         //Serial.println(" alarm");
20         digitalWrite(buzzer,HIGH);
21         delay(500);
22         digitalWrite(buzzer,LOW);
23         delay(500);
24         brojac++;
25         if(brojac>5)break;
26     }
27     brojac = 0;
28     flags(0, 1, 0);
29     flagKontrolni = 0;
30     flagAlkohol = 1;
31     //Serial.println("alkohol visok");
32
33 }
34 else {
35     // ukoliko je sve uredi, pali se motor
36     flags(1, 0, 1);
37     flagKontrolni = 1;
38     flagAlkohol = 0;
39 }
40 }
41
42 else {
43     // Serial.println("NEMA PRISTUPA");
44     flags(0, 1, 0);
45     flagKontrolni = 0;
46
47     delay(300);
48 }
49 if (digitalRead(kljuc) == HIGH && flagAlkohol == 1) {
50     flagAlkohol = 0;
51
52 }
53
54 }

```

Naredna funkcija implementira kod za pokretanje RFID čitanja. Prvo se čisti trenutna identifikaciona riječ, te se funkcijom članicom "PICCReadCardSerial()" mfrc522 instance čita identifikaciona riječ kartice. U ovisnosti od pročitane informacije se spajanjem karaktera dobija tražena riječ, čije se vrijednosti slova radi lakšeg upoređivanja pretvara u velika.

## Program 3.9: Funkcija za kontrolu RFID čitača

```

1 // kupljenje podataka sa citaca RFID
2 void citanjeKartice(){
3     content="";
4     mfrc522.PICC_ReadCardSerial(); // citanje RFID citacem
5     byte letter;

```

```
6   for (byte i = 0; i < mfrc522.uid.size; i++) {
7       content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "_0" : "_"
8           ));
9       content.concat(String(mfrc522.uid.uidByte[i], HEX));//kreiranje
10          UID rijeci
11   }
12   content.toUpperCase(); // prebacivanje rijeci u sva velika slova
13 }
```

Zadnja funkcija prve cjeline koda služi za vizualni prikaz stanja propusnosti veze motor i akumulator. U ovisnosti od vrijednosti predanih parametara se aktiviraju signalne, svijetleće diode

#### Program 3.10: Funkcija za kontrolu zastavica

```
1  // Signalne zastavice
2  void flags(int a, int b, int c) {
3      if (a == 0) {
4          digitalWrite(pass, LOW);
5      } else {
6          digitalWrite(pass, HIGH);
7      }
8      if (b == 0) {
9          digitalWrite(block, LOW);
10     } else {
11         digitalWrite(block, HIGH);
12     }
13     if (c == 0) {
14         digitalWrite(relay, LOW);
15     } else {
16         digitalWrite(relay, HIGH);
17     }
18 }
```

Dalje će se opisati kod za reguliranje rada druge cjeline sistema. Ova cjelina, kako je rečeno prethodnim poglavljem, vrši komunikaciju sa GPS i GSM modulom, te se preko nje implementira logika za upravljanje propusnosti putem SMS poruke.

Prvi isječak koda vrši uvođenje potrebnih biblioteka za lakšu interakciju sa GPS modulom, te za kreiranje softverski definiranih pinova za serijsku komunikaciju. Potom se vrši kreiranje instanci uvedenih klasa.

#### Program 3.11: Inicijalizacija za drugu cjelinu

```
1  #include <SoftwareSerial.h>
2  #include <TinyGPS++.h>
3
4  //deginiranje softverskih serijskih ponova za komunikaciju
5  SoftwareSerial mySerial(9, 10);
6  SoftwareSerial ss(4, 3);
7  // The TinyGPS++ objekt, za pogodan nacin dobijanja parsiranih GPS
8  podataka
9  TinyGPSPlus gps; digitalWrite(relay, HIGH);
10 }
```

U "setup()" funkciji se vrši inicijalizacija serijske komunikacije za hardverske i softverske pinove. Definirane su varijable za primanje GPS pozicije, i SMS poruke. Upravljanje radom

GSM modula se vrši preko definiranih komandnih riječi, od koji "AT+IPR=0" postavlja modul u tekstualni način rada.

**Program 3.12:** Postavna funkcija za drugu cjelinu

```

1 void setup()
2 {
3   mySerial.begin(9600);
4   Serial.begin(9600);
5   delay(100);
6   //mySerial.println("AT+IPR=0");    //Postavljanje GSM modula u
      tekst mode
7   // Definiranje stringova za prikupljanje odgovora od modula, GPS i
      GSM
8   String a=""; // za spremanje sadržaja SMS poruke
9   String GPSposition=""; // za spremanje GPS pozicije
10  int timer;
11  RecieveMessage(); // funkcija za omogućavanje primanja SMS poruke
12 }
13 }
```

U iterativnoj funkciji se prvo omogućava GSM modulu primanje poruke putem "RecieveMessage()" funkcije. Ukoliko je prisutna serijska informacija sa modula, ista se preuzima u varijablu "a". Ukoliko je poruka primljena sa navedenog broja, njen sadržaj se obrađuje. U ovisnosti od sadržaja poruke, serijskim putem se šalje tekst prvoj Arduino pločici, koja na osnovu toga mijenja stanje propusnosti motora i aktuatora. Ukoliko SMS poruka sadrži riječ "Lokacija", kupi se GPS podatak koji se parsira na pogodan način, i šalje povratnom SMS porukom pošiljaocu. Povratni SMS se šalje putem "SendMessage(sadržaj)" funkcije.

**Program 3.13:** Iterativna funkcija za drugu cjelinu

```

1 void loop()
2 {
3   RecieveMessage();
4
5   if (mySerial.available()>0){ // Ukoliko GPS modul ima podataka za
      poslati
6     a = mySerial.readString();
7     //Serial.println(a);
8     if(a.indexOf("38761295746") > 0){ // Ukoliko poruka dolazi od
      specifikiranog broja
9       if(a.indexOf("Ugasi") > 0){ // Fje u ovisnosti od teksta
      poruke
10        Serial.println("-iskljuci");
11      }
12      if(a.indexOf("Upali") > 0){
13        Serial.println("-ukljuci");
14      }
15      if(a.indexOf("Lokacija") > 0){
16        ss.begin(9600);
17        delay(1000);
18        timer = millis();
19        while(!ss.available() && (millis()
20          -timer)<5000){}
21        if (ss.available() > 0){
22          gps.encode(ss.read());
23          if (gps.location.isUpdated()){//Kupeljenje podataka sa
      GPS modula, i njihov pogodan prikaz za SMS poruku
24            GPSposition="";
```

```
25         GPSPosition.concat("latitude=");
26         GPSPosition.concat(String(gps.location.lat()));
27         GPSPosition.concat("_longitude=");
28         GPSPosition.concat(String(gps.location.lng()));
29
30     }
31     }else{
32         GPSPosition="Nema_signala";
33     }
34     SendMessage(GPSPosition);
35     ss.end();
36     Serial.println("-lokacija");
37 }
38 }
39
40 }
41 }
```

Sljedeći isječak prikazuje funkcije za slanje i primanje SMS poruka, putem GPS modula. Kao što je rečeno, modulom se upravlja preko kratkih naredbi koje započinju sadržajem "AT". "SendMessage(sadržaj)" funkcija prima kao parametar sadržaj poruke, u ovom slučaju GPS lokacija vozila.

**Program 3.14:** Funkcija za slanje i primanje SMS poruke

```
1  void SendMessage(String poruka)
2  {
3      mySerial.println("AT+CMGF=1");
4      delay(1000);
5      mySerial.println("AT+CMGS=\"+38761295746\"\r"); // Definiranje
        broja na koji se salje SMS
6      delay(1000);
7      mySerial.println(poruka); // Definiranje poruke za slanje
8      delay(100);
9      mySerial.println((char)26);
10     delay(1000);
11 }
12
13
14 void RecieveMessage()
15 {
16     mySerial.println("AT+CNMI=2,2,0,0,0"); // Naredba za primanje SMS
        poruka u realno vrijeme
17     delay(1000);
18 }
```

## 3.4 Blokovski kod

Mobilna aplikacija, koju službenik samostalno pokreće, je kreirana Mit App Inventor aplikacijom. Proces razvijanja koda je vizualan, svaki blok nosi svoju funkciju.

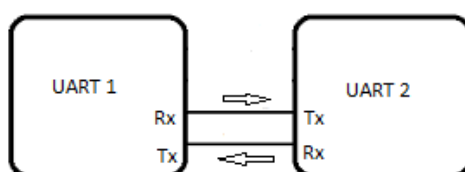
Prvi grupa blokova, prikazana na slici 3.4., omogućava odabiranje bluetooth uređaja kojim će aplikacija komunicirati, sa liste uređaja sa kojima je moguće ostvariti taj tip komunikacije. Svaki uređaj je prikazan putem svoje adrese, i naziva.





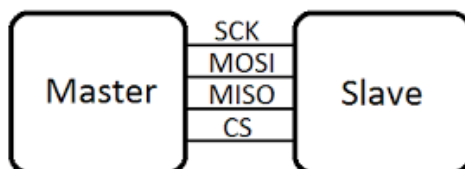
visoko, te je prikazano putem napona kao približno 0, ili 5 V DC. Komunikacija sa alkolnim sensorom je ostvarena putem čitanja analogne vrijednosti sa "AOUT" izlaza. Analogni ulaz je u stanju registrovati 1024 nivoa vrijednosti.

Većina ostalih modula, tj. GPS, GSM i Bluetooth moduli, kao i sama komunikacija među dvjema Arduino Uno pločama, je ostvarena putem UART modula. Ova komunikacija je asinhronog tipa, te zahtjeva samo dva kanala za ostvarenje razmjene podataka. UART modul vrši pretvaranje paralelnog prikaza podataka, dobijenog čitanjem registra, u serijski prikaz u vidu niza bitova, spremnih za slane. Grafički prikaz serijske komunikacije je prikazan na slici 3.7.



**Slika 3.7:** Prikaz UART komunikacije.

Na samom kraju, komunikacija Arduina Una sa RFID čitačem je ostarena korištenjem SPI interfejsa. Ovaj interfejs je sinhronog tipa, i koristi četiri komunikacione linije za brži prijenos podataka. Dvije linije prenose podatke, dok su ostale upravljačke i služe za odabiranje čipa sa kojim se želi komunicirati, kao i za slanje takta za sinhronizaciju. Grafički prikaz interfejsa je vidljiv na slici 3.8.



**Slika 3.8:** Prikaz SPI komunikacije.

Ovim su opisani svi tipovi komunikacije među modulima unutar ovog sistema.

## Poglavlje 4

### Eksperimentalna analiza rada sistema

Sistem se ugrađuje kao dodatak na već postojeću mehanizaciju auta, ali može biti i ugrađen kao početni mehanizam. U početnom stanju je propusnost blokirana, jer ne dopušta vezu akumulatora sa motorom.



**Slika 4.1:** Prikaz sistema u stanju blokiranja.

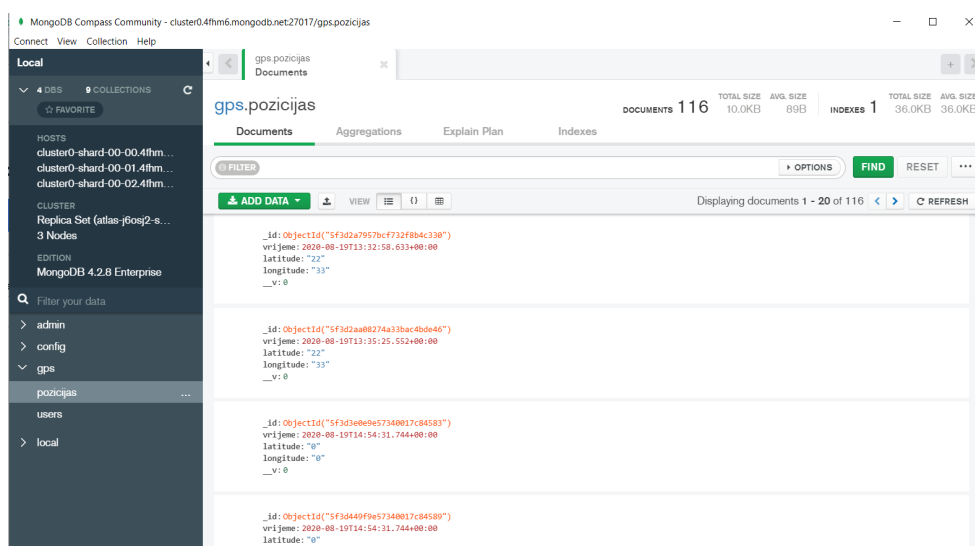
Stanje blokiranja je signalizirano crvenom diodom, kao što je vidljivo na slici 4.1. Ukoliko tokom vožnje dođe do registriranja alkoholnih para, pokreće se alarm preko zujalice, te se nakon određenog vremena auto gasi. Dobija prikaz kao na slici 4.1. Ukoliko se desi pozitivan rezultat procesa autorizacije, pali se plava dioda, te je motor priključeno na akumulator. Sistem u stanju vođenja je prikazan na slici 4.2.



**Slika 4.2:** Prikaz sistema u dopuštenom stanju.

Ako sistem, preko MQ3 senzora, detektira nedozvoljenu količinu alkohola u zraku, pokreće se zvučna signalizacija. Nakon kratkog vremenskog intervala, dostatnog za vozača da se skloni iz saobraćaja, vozilo se gasi.

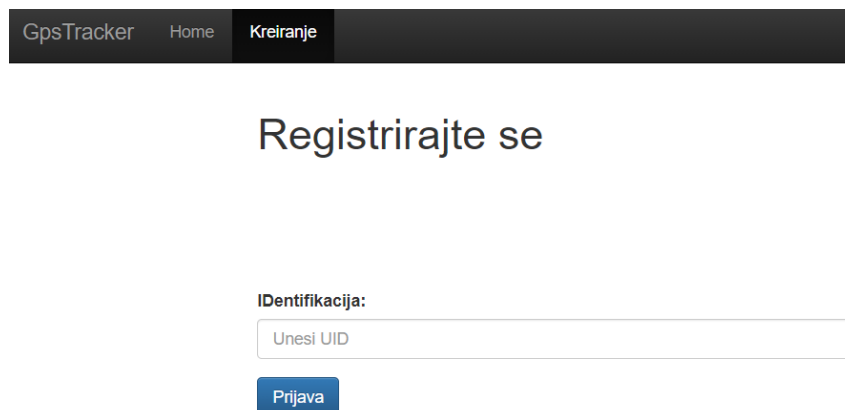
Nakon uspješne prijave vozač pokreće aplikaciju, te odabire bluetooth predajnik sistema, preko dugmeta "Odaberi uređaj". Tad dolazi do slanja GPS podataka na server, čija je adresa kodirana u aplikaciju. Prikaz primljenih podataka je vidljiv na online prikazu baze podataka, korištenjem servisa "Atlas", na slici 4.3.



**Slika 4.3:** Prikaz baze podataka sa GPS podatcima.

U cilju prikaza podataka, kreirana je pomenuta internet aplikacija, korištenjem razvojnog okvira "Express". Ulaskom na stranicu vlasnik kreira naziv korisnika od kojeg želi primati in-

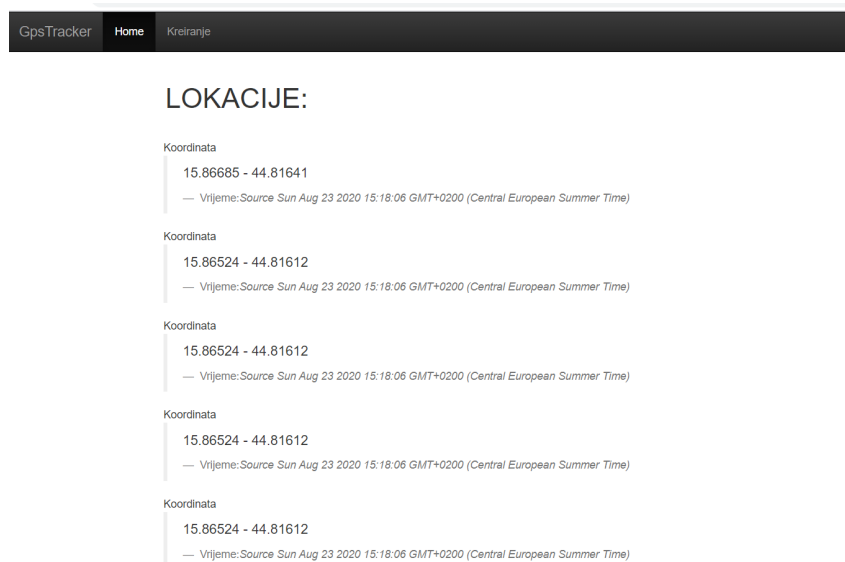
formacije o poziciji. Ista identifikacija je dana korisniku za unos u svoj uređaj. Prikaz kreiranja korisnika je vidljiv na slici 4.4.



The screenshot shows the 'Kreiranje' (Creation) tab selected in the top navigation bar. The main heading is 'Registrirajte se' (Register). Below it, there is a section titled 'Identifikacija:' with a text input field labeled 'Unesi UID' (Enter UID) and a blue 'Prijava' (Login) button.

**Slika 4.4:** Prikaz prozora za registraciju.

Nakon slanja podataka, pristupnik stranice moguće vidjeti sve GPS podatke koje je auto ostvarilo od izlaska iz stanice, do vraćanja u nju. Prikaz podataka je vidljiv na slici 4.18.



The screenshot shows the 'Home' tab selected in the top navigation bar. The main heading is 'LOKACIJE:' (LOCATIONS:). Below it, there are five identical entries, each showing coordinates and a timestamp. The data is as follows:

Koordinata	Vrijeme: Source
15.86685 - 44.81641	Sun Aug 23 2020 15:18:06 GMT+0200 (Central European Summer Time)
15.86524 - 44.81612	Sun Aug 23 2020 15:18:06 GMT+0200 (Central European Summer Time)
15.86524 - 44.81612	Sun Aug 23 2020 15:18:06 GMT+0200 (Central European Summer Time)
15.86524 - 44.81612	Sun Aug 23 2020 15:18:06 GMT+0200 (Central European Summer Time)
15.86524 - 44.81612	Sun Aug 23 2020 15:18:06 GMT+0200 (Central European Summer Time)

**Slika 4.5:** Prikaz GPS podataka na stranici.

Ukoliko dođe do krađe automobila, moguće je isto zaustaviti slanjem SMS poruke, sadržaja "Ugasi", te se dobija prikaz kao na slici 4.1. Nakon toga nije moguće ponovo upaliti vozilo, dok se ne primi poruka sa sadržajem "Upali". Dodatna funkcija podsistema je kupljenje GPS lokacije, u vidu povratne SMS poruke, slanjem poruke sadržaja "Lokacija". Ovaj mehanizam je osmišljen kao pomoćni dodatak pri pronalaženju vozila u slučaju krađe.

## Poglavlje 5

### Zaključak

Kao što je uvodom rečeno, ovaj rad je prikazati postupak kreiranja sistema za onemogućavanje paljenja, i kontroliranja, vozila od strane osoba u alkoholiranom stanju. Stanje se registrira senzorom za mjerenje alkoholnih para. Pošto je sistem namjenjen za kontrolu službenih vozila, osobe moraju proći proces autorizacije. Pored ovih funkcija, sistem šalje serveru podatke o GPS lokaciji.

Alkoholne pare se registriraju MQ3 senzorom, koji mjeri prisutni etanol u zraku. Ukoliko je osoba prošla proces autorizacije, korištenjem RFID identifikacionih kartica, i usljed nepostojanja alkoholnih para, pali se vozilo. Aukumulator se spaja na motor vozila preko releja, signalom sa Arduino Uno pločice. Svakih deset sekundi se serveru šalje informacija o trenutnoj poziciji vozila. Ukoliko je auto ukradeno, postoji funkcija slanja SMS poruke, sa riječi "Ugasi", koja omogućuje gašenje vozila preko GSM modula. Ova funkcija onemogućuje daljnje korištenjem vozila od strane ukradenog.

Iako je sistem neidealno, što je vidljivo kroz omogućen pristup unutrašnjem mehanizmu od strane osoba unutar auta jer je oklop sačinjen od kartona, te slaba tačnost identifikacije alkoholisanog stanja osobe koja upravlja vozilom, projektom je dao solidan mehanizam koji, prije svega demonstrativno, pokazuje način onemogućavanja upravljanja vozilom u opasnom stanju.

# **Prilozi**

# Prilog A

## Kodni program projekta

Ovaj prilog sadrži cjelokupni kod projekta, kako Arduino dio za obje cjeline.

### A.1 Arduino kod

Kod, za prvu cjelinu, je:

**Program A.1:** Cjelokupni kod za prvu cjelinu

```
1
2 // Biblioteka Softverski Serijske pinove; za RFID citac
3 #include <SPI.h>
4 //#include <avr/wdt.h>
5 #include <MFRC522.h>
6 //definiranje ulaza/izlaza za RFID citac
7 #define SS_PIN 10
8 #define RST_PIN 9
9 MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
10
11 int AOUTpin = 0; //analogni izlaz MQ3 senzora
12 int DOUTpin = 8; //digitalni izlaz MQ3 senzora
13 int pass = 7; //signalni izlaz za odobrenje paljenja
14 int block = 6; //signalni izlaz za onemogucavanje paljenja
15 int relay = 5; //signalni izlaz za upravljacku elektroda MOSFETA
16 int kljuc = 4; // ulaz za signaliziranje zeljenog paljenja
17 int buzzer = 3; // izlaz za paljenje zujalice
18
19 // kontrolne zastavice i tajmeri
20 int flagKontrolni = 0; // za omogucavanje nesmetanog rada propusne
    veze akumulator-motor
21 int flagAlkohol = 0; // zastavica za
22 int brojac = 0; // koristen za mjerenja broja zvucnih
    signalizacija
23 float timer1 = 0; // koristen za slanje GPS poticaja
24 int GSMflag = 1; // upravljanje propusnim stanjem putem SMS poruke
25
26 String content = ""; //Varijabla za spremanje identifikacione rijeci
27 String SMSporuka="";
28 //varijabla za spremanje nivoa sa alkoholnog senzora
29 int limit;
30
31
32
```

```
33 void setup()
34 {
35     Serial.begin(9600);    // pokretanje serijske komunikacije
36     SPI.begin();           // pokretanje softverske serijske veze
37     mfrc522.PCD_Init();    // pokretanje RFID citaca
38     //definiranje ulaza/izlaza
39     pinMode(DOUTpin, INPUT);
40     pinMode(pass, OUTPUT);
41     pinMode(block, OUTPUT);
42     pinMode(relay, OUTPUT);
43     pinMode(buzzer, OUTPUT);
44     pinMode(kljuc, INPUT);
45     digitalWrite(buzzer, LOW); // onemogucavanje zujalice
46     flags(0, 1, 0); //inicijalizacija signlanih svjetala
47     timer1 = millis();
48 }
49
50
51 void loop() {
52
53     //blok za kontrolu stanja putem SMS poruke
54     if (Serial.available() > 0) {
55         SMSporuka = Serial.readString();
56         if (SMSporuka.indexOf("iskljuci") > 0) {
57             GSMflag = 0;
58             SMSporuka = "";
59
60         }
61         if (SMSporuka.indexOf("ukljuci") > 0) {
62             GSMflag = 1;
63             SMSporuka = "";
64         }
65     }
66
67
68     // Slanje impulsa uredjaju za kupljenje GPS podataka, i slanje
        istih
69     if (millis() - timer1 > 10000) {
70         Serial.println("1");
71         timer1 = millis();
72     }
73
74     //pozivanje citaca, ukoliko je nova kartica prisutna
75     if (mfrc522.PICC_IsNewCardPresent()) {
76         citanjeKartice();
77     }
78
79
80
81     //odobravanje pokretanja ukoliko je tacna identifikacija i zeljeno
        paljenje vozila, ili ako je vec jednom odobreno tokom iste
        voznje -
82     // glavni uvjet je nepostojanje alkoholnih para u svakom trenutku
        voznje
83
84
85
86     if (((content=="_33_87_3B_39" && digitalRead(kljuc) == LOW) || (
```



```
digitalRead(kljuc) == LOW && flagKontrolni == 1)) &&
flagAlkohol == 0 && GSMflag==1){
87   flagKontrolni = 1;
88   for (uint8_t i = 0; i < mfrc522.uid.size; i++) { // brisanje
        buffera RFID citaca
89   mfrc522.uid.uidByte[i] = 0;
90   content="";
91 }
92 //Serial.println("OTVORENO");
93 value = analogRead(AOUTpin); //dobijanje izlaza mq3 senzora
94 delay(100);
95 // ukoliko se iskaze zelja gasenja, isto se gasi
96 if (digitalRead(kljuc) == HIGH) {
97     flagKontrolni = 0;
98     flags(0, 1, 0);
99 }
100 // ukoli je kolicina alkoholne pare veca od nivoaa, pokrene se
    alarm zujalicom, i nakon odredenog vremena se auto gasi
101 if (value > 857) {
102     while(true){
103         //Serial.print(brojac);
104         //Serial.println(" alarm");
105         digitalWrite(buzzer,HIGH);
106         delay(500);
107         digitalWrite(buzzer,LOW);
108         delay(500);
109         brojac++;
110         if(brojac>5)break;
111     }
112     brojac = 0;
113     flags(0, 1, 0);
114     flagKontrolni = 0;
115     flagAlkohol = 1;
116     //Serial.println("alkohol visok");
117
118 }
119 else {
120     // ukoliko je sve ureda, pali se motor
121     flags(1, 0, 1);
122     flagKontrolni = 1;
123     flagAlkohol = 0;
124 }
125 }
126
127 else {
128     // Serial.println("NEMA PRISTUPA");
129     flags(0, 1, 0);
130     flagKontrolni = 0;
131
132     delay(300);
133 }
134 if (digitalRead(kljuc) == HIGH && flagAlkohol == 1) {
135     flagAlkohol = 0;
136
137 }
138
139 }
140
```

```

141
142 //////////////////////////////////////////////////
143 // kupljenje podataka sa citaca RFID
144 void citanjeKartice(){
145 content="";
146 mfrc522.PICC_ReadCardSerial(); // citanje RFID citacem
147 byte letter;
148 for (byte i = 0; i < mfrc522.uid.size; i++) {
149     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "_0" : "_"
150         ));
151     content.concat(String(mfrc522.uid.uidByte[i], HEX)); //kreiranje
152         UID rijeci
153 }
154 content.toUpperCase(); // prebacivanje rijeci u sva velika slova
155 }
156
157 //////////////////////////////////////////////////
158 // Signalne zastavice
159 void flags(int a, int b, int c) {
160     if (a == 0) {
161         digitalWrite(pass, LOW);
162     } else {
163         digitalWrite(pass, HIGH);
164     }
165     if (b == 0) {
166         digitalWrite(block, LOW);
167     } else {
168         digitalWrite(block, HIGH);
169     }
170     if (c == 0) {
171         digitalWrite(relay, LOW);
172     } else {
173         digitalWrite(relay, HIGH);
174     }
175 }

```

Kod, za drugu cjelinu, je:

#### Program A.2: Cjalokupni kod za drugu cjelinu

```

1  #include <SoftwareSerial.h>
2  #include <TinyGPS++.h>
3
4  //deginiranje softverskih serijskih ponova za komunikaciju
5  SoftwareSerial mySerial(9, 10);
6  SoftwareSerial ss(4, 3);
7  // The TinyGPS++ objekt, za pogodan nacin dobijanja parsiranih GPS
8  // podataka
9  TinyGPSPPlus gps;
10
11 void setup()
12 {
13     mySerial.begin(9600);
14     Serial.begin(9600);
15     delay(100);
16     //mySerial.println("AT+IPR=0"); //Postavljanje GSM modula u
17         tekst mode
18     // Definiranje stringova za prikupljanje odgovora od modula, GPS i

```

```

    GSM
17   String a=""; // za spremanje sadrzaja SMS poruke
18   String GPSposition=""; // za spremanje GPS pozicije
19   int timer;
20   RecieveMessage(); // funkcija za omogucavanje primanja SMS poruke
21 }
22
23
24 void loop()
25 {
26   RecieveMessage();
27
28   if (mySerial.available()>0){ // Ukoliko GPS modul ima podataka za
        poslati
29     a = mySerial.readString();
30     //Serial.println(a);
31     if(a.indexOf("38761295746") > 0){ // Ukoliko poruka dolazi od
        specifikiranog broja
32       if(a.indexOf("Ugasi") > 0){ // Fje u ovisnosti od teksta
        poruke
33         Serial.println("-iskljuci");
34       }
35       if(a.indexOf("Upali") > 0){
36         Serial.println("-ukljuci");
37       }
38       if(a.indexOf("Lokacija") > 0){
39         ss.begin(9600);
40         delay(1000);
41         timer = millis();
42         while(!ss.available() && (millis()
43           -timer)<5000){}
44         if (ss.available() > 0){
45           gps.encode(ss.read());
46           if (gps.location.isUpdated()){//Kupeljenje podataka sa
        GPS modula, i njihov pogodan prikaz za SMS poruku
47             GPSposition="";
48             GPSposition.concat("latitude=");
49             GPSposition.concat(String(gps.location.lat()));
50             GPSposition.concat("_longitude=");
51             GPSposition.concat(String(gps.location.lng()));
52
53           }
54           }else{
55             GPSposition="Nema_signala";
56           }
57           SendMessage(GPSposition);
58           ss.end();
59           Serial.println("-lokacija");
60         }
61       }
62
63     }
64   }
65
66
67 void SendMessage(String poruka)
68 {
69   mySerial.println("AT+CMGF=1");
```

```
70     delay(1000);
71     mySerial.println("AT+CMGS=\"+38761295746\"\r"); // Definiranje
        broja na koji se salje SMS
72     delay(1000);
73     mySerial.println(poruka); // Definiranje poruke za slanje
74     delay(100);
75     mySerial.println((char)26);
76     delay(1000);
77 }
78
79
80 void RecieveMessage()
81 {
82     mySerial.println("AT+CNMI=2,2,0,0,0"); // Poruka za primanje SMS
        poruka u realno vrijeme
83     delay(1000);
84 }
```

# Literatura

- [1] Tawil, Y., “Understanding arduino uno hardware design”, dostupno na: <https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/> (05. august 2020.).
- [2] Garmin, “What is gps”, dostupno na: <https://www.garmin.com/en-US/AboutGPS/> (08. august 2020.).
- [3] Group, C., “Hc-05 - bluetooth module”, dostupno na: <https://components101.com/wireless/hc-05-bluetooth-module> (06. august 2020.).
- [4] Group, C., “Active pasive buzzer”, dostupno na: <https://components101.com/buzzer-pinout-working-datasheet> (06. august 2020.).
- [5] Group, C., “Mq-3 alcohol gas sensor”, dostupno na: <https://components101.com/sensors/mq-3-alcohol-gas-sensor> (06. august 2020.).
- [6] Group, C., “Rc522 rfid module”, dostupno na: <https://components101.com/wireless/rc522-rfid-module> (06. august 2020.).
- [7] Woodford, C., “Relays”, dostupno na: <https://www.explainthatstuff.com/howrelayswork.html> (06. august 2020.).