

Raskrižje realizirano pomoću Arduina s prednošću prolaska za žurna vozila

Batinac, Alan

Undergraduate thesis / Završni rad

2023

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:187:564877>

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

Download date / Datum preuzimanja: **2024-07-11**



Sveučilište u Rijeci, Pomorski fakultet
University of Rijeka, Faculty of Maritime Studies

Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Maritime Studies - FMSRI Repository](#)



uniri DIGITALNA
KNJIŽNICA



SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET

ALAN BATINAC

RASKRIŽJE REALIZIRANO POMOĆU ARDUINA S
PREDNOŠĆU PROLASKA ZA ŽURNA VOZILA

ZAVRŠNI RAD

Rijeka, 2023.

SVEUČILIŠTE U RIJECI

POMORSKI FAKULTET

**RASKRIŽJE REALIZIRANO POMOĆU ARDUINA S
PREDNOŠĆU PROLASKA ZA ŽURNA VOZILA**

**AN INTERSECTION REALIZED WITH THE HELP OF
ARDUINO WITH THE ADVANTAGE OF PASSING FOR
EMERGENCY VEHICLES**

ZAVRŠNI RAD

Kolegij: Inteligentni transportni sustavi

Mentor: izv. prof. dr. sc. Jasmin Čelić

Student: Alan Batinac

Studijski smjer: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 01120760272

Rijeka, kolovoz 2023

Student: Alan Batinac

Studijski program: Elektroničke i informatičke tehnologije u pomorstvu

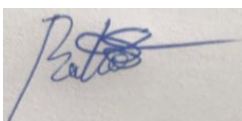
JMBAG: 01120760272

IZJAVA O SAMOSTALNOJ IZRADI ZAVRŠNOG RADA

Kojom izjavljujem da sam završni rad s naslovom **Raskrižje realizirano pomoću Arduina s prednošću prolaska za žurna vozila** izradio samostalno pod mentorstvom izv. prof. dr. sc. Jasmina Čelića.

U radu sam primijenio/la metodologiju izrade stručnog/znanstvenog rada i koristio/la literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezao s fusnotama i korištenim bibliografskim jedinicama, te nijedan dio rada ne krši bilo čija autorska prava. Rad je pisan u duhu hrvatskoga jezika.

Student:



Alan Batinac

Student: Alan Batinac

Studijski program: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 01120760272

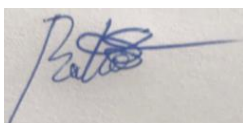
IZJAVA STUDENTA – AUTORA

O JAVNOJ OBJAVI OBRANJENOG ZAVRŠNOG RADA

Izjavljujem da kao student – autor završnog rada dozvoljavam Pomorskom fakultetu Sveučilišta u Rijeci da ga trajno javno objavi i besplatno učini dostupnim javnosti u cjelovitom tekstu u mrežnom digitalnom repozitoriju Pomorskog fakulteta.

U svrhu podržavanja otvorenog pristupa završnim radovima trajno objavljenim u javno dostupnom digitalnom repozitoriju Pomorskog fakulteta, ovom izjavom dajem neisključivo imovinsko pravo iskorištavanja bez sadržajnog, vremenskog i prostornog ograničenja mog završnog rada kao autorskog djela pod uvjetima *Creative Commons* licencije CC BY Imenovanje, prema opisu dostupnom na <http://creativecommons.org/licenses/>

Student – autor



Alan Batinac

SAŽETAK

U ovom završnom radu rađena je maketa pametnog raskrižja pomoću Arduina Una. Kroz rad se istražuje poboljšanje funkcionalnosti raskrižja s naglaskom na osiguravanje prednosti prolaska hitnim vozilima. Kroz dinamičko upravljanje semaforima i senzorske sustave, raskrižje se prilagođava situacijama hitne intervencije, omogućavajući brži prolazak i time povećavajući sigurnost i učinkovitost u urbanom prometnom okruženju. Ovaj pristup ima potencijal značajno smanjiti reakcijsko vrijeme hitnih službi i stvoriti bolje uvjete za promet.

Ključne riječi: Arduino, mikrokontroleri, svjetleće diode, RFID senzori

SUMMARY

In this final work, a model of a smart intersection was made using Arduino Uno. The work investigates the improvement of the functionality of the intersection with an emphasis on ensuring the advantage of passing by emergency vehicles. Through dynamic management of traffic lights and sensor systems, the intersection adapts to emergency situations, enabling faster passage and thus increasing safety and efficiency in an urban traffic environment. This approach has the potential to significantly reduce the response time of emergency services and create better traffic conditions.

Keywords: Arduino, microcontrollers, LED diodes, RFID sensors

SADRŽAJ

SAŽETAK.....	III
SUMMARY	III
SADRŽAJ	IV
1. UVOD	1
2. ARDUINO.....	2
2.1 POVIJEST ARDUINA.....	2
2.2 PRAKTIČNE PRIMJENE.....	3
2.3 ARDUINO PROGRAM	3
2.4 FUNKCIJE	5
3. DIJELOVI ARDUINA	6
3.1 DIGITALNI ULAZI/IZLAZI	7
3.2 ANALOGNI ULAZI/IZLAZI	7
3.3 NAPAJANJE	8
3.4 ICSP ULAZI/IZLAZI.....	8
3.5 ATmega328	8
3.6 PREOSTALI DJELOVI	9
4. SENZORI.....	10
4.1 RFID SENZOR RC522	11
4.2 IMPLEMENTACIJA RFID SENZORA U RASKRIŽJE	13
5. SVJETLEĆE DIODE.....	14
6. KOMUNIKACIJA IZMEĐU 2 ARDUINA	16
7. PAMETNO RASKRIŽJE	18
8. IZRADA RADA.....	20
8.1 PRIKAZ KODA NA MASTER ARDUINU (ARDUINO UNO)	24
8.2 PRIKAZ KODA NA SLAVE ARDUINU (ARDUINO MEGA).....	27
9. ZAKLJUČAK.....	32
LITERATURA	33
KAZALO KRATICA	35

1. UVOD

U današnje vrijeme tehnologija sve više napreduje i postaje sve više dostupna te olakšava svakodnevni život. U posljednjem desetljeću Arduino platforma se razvila i postala popularna za izradu mnoštva projekata i razvoj raznih tehnoloških rješenja u području informatike i elektronike. U modernom urbanom okruženju, prometna raskrižja predstavljaju ključne čvorove u cestovnoj infrastrukturi. Učinkovito upravljanje raskrižjima igra ključnu ulogu u optimizaciji prometnog toka i povećanju sigurnosti sudionika u prometu. Jedan od izazova s kojima se suočavaju graditelji i planeri prometnih sustava jest osigurati brz i siguran prolaz hitnim vozilima, poput hitnih medicinskih vozila ili vatrogasnih kamiona, kako bi pravovremeno pružili pomoć i spašavanje u hitnim situacijama.

U svrhu rješavanja ovog izazova, sve više se koristi napredna tehnologija, a jedno od inovativnih rješenja je implementacija Arduino platforme za inteligentno upravljanje raskrižjima.

Arduino je open-source platforma pomoću koje se može eksperimentirati, stvarati i razvijati prototipe. Postoje više vrsta programskih pločica kao što su: Arduino Uno, Arduino Mega, Arduino Nano i mnoge druge. Prednosti Arduina su dostupnost i cijena, te način komunikacije između korisnika i sustava. Programski jezik koji se koristi je C/C++, te se pisani programi izvršavaju korak po korak.

U praktičnom djelu rada izrađen je projekt raskrižja koji ima prednost za žurna vozila te je realiziran pomoću RIFD senzora.

Programske pločice imaju mogućnost komunikacije sa različitim vrstama senzora gdje primaju signale i šalju ih dalje.

2. ARDUINO

Arduino je vrlo jednostavan open-source mikrokontroler koji omogućuje korisnicima da eksperimentiraju u području elektronike i programiranja. Arduino platforma omogućuje korisnicima da razvijaju prototipe, eksperimentiraju i stvaraju interaktivne uređaje. Arduino je postao popularan radi svoje jednostavnosti kod korištenja kao i kod jednostavnosti samog programiranja čime je postao pristupačan i ostalim korisnicima koji nemaju prethodnog znanja. Arduino mikrokontroler se sastoji od analogno-digitalnih ulaza/izlaza, mogućnosti povezivanja sa raznim sensorima i drugim električnim komponentama i mikrokontrolerima.

Za razvoj i samo programiranje se koristi C/C++ programski jezik za pisanje i programiranje mikrokontrolera. Pogodan je za različite aspekte elektronike kao što su osvjetljenje, motori, senzori, displeji itd. Mogu se izrađivati razni projekti poput automatiziranja kućanstva, pametnih uređaja, robotike i umjetnosti.

Neki najpoznatiji uređaji su Arduino Uno, Arduino Mega i Arduino Nano koji koriste isti program kod programiranja. Jedna od prednosti Arduina je zajednica korisnika koji dijele svoja iskustva, znanja i rješenja za neke probleme, te nude i mnoštvo resursa.

2.1 POVIJEST ARDUINA

Arduino projekt je započeo 2005. godine u Iveri, Italiji, kada su Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino i David Mellis odlučili stvoriti jednostavan i pristupačan alat za programiranje i razvoj. Ideja je bila razviti platformu koja bi omogućila ljudima bez tehničkog znanja da kroz zabavnu interakciju uđu u svijet elektronike i programiranja. Prva Arduino ploča nazvana Arduino NG (New Generation) predstavljena je javnosti 2005. godine i bila je bazirana na mikrokontroleru Atmel ATmeg168 koji je imao USB sučelje za jednostavnost programiranja i prenošenje koda.

Jedan od aspekata koji je proslavio Arduino je bila odluka da platforma bude open-source što znači da su svi izvorni kodovi, sheme i dizajni svima dostupni. Zbog tog razloga Arduino platforma je postala jako popularna među studentima, hobistima, umjetnicima i inženjerima. Korisnici su na jednostavan način razvijali razne prototipe i stvarali nove inovativne projekte u raznim područjima robotike, automatizacije kućanstva itd.

2.2 PRAKTIČNE PRIMJENE

Kako je Arduino postao popularan zbog svoje jednostavnosti korištenja, koristi se u raznim područjima. Danas se Arduino najviše koristi pri izradi prototipa električnih uređaja jer omogućuje brzo i jednostavno testiranje ideja prije prelaska na kompliciranije i složenije modele. Mogućnost jednostavnog i brzog povezivanja s različitim sensorima i komponentama korisnicima nudi jednostavnu izradu svakakvih raznih ideja i prototipa.

Također je velika primjena Arduina u pametnim kućama gdje nudi alternativna rješenja, a zbog svoje jednostavnosti uporabe korisnici sami mogu automatizirati svoje domove. Neki od primjera su automatizacija svjetla i elektroničkih uređaja, kontrola temperature i padalina. Korisnik ima mogućnost sam odrediti raspored paljenja i gašenja svjetla ili prilagoditi osvjetljenje prema uvjetima okoline. Uz navedeno može se dodati i sigurnost doma i praćenje pomoću senzora za pokret, senzora otvaranja vrata i prozora. Sustav može obavijestiti vlasnika porukom ako se primijeti neobična aktivnost.

Arduino se može koristiti i u poljoprivredi kako bi korisnicima pomogao pri praćenju rasta biljaka, automatizaciji sustava za navodnjavanje i mjerenju vlažnosti tla i zraka.

2.3 ARDUINO PROGRAM

Arduino koristi različite programske alate za razvoj i programiranje. Najčešće korišten program je Arduino IDE (Integrated Development Environment). Arduino IDE je besplatan softverski program koji korisnicima pruža sve što im je potrebno za pisanje, uređivanje, prevođenje i prenošenje koda na Arduino pločicu. Program je dostupan za Windows, Mac i Linux operativne sustave, te je jednostavan za instalaciju. Svaki Arduino kod mora imati dvije osnovne petlje koje su `void setup{}` i `void loop{}`. U `setup` funkciji se definiraju parametri koje želimo očitati, varijable, ili za postavu pinova kao ulazni ili izlazni. U funkciji `loop` se izvršava petlja sve dok Arduino ima napajanje, te je ona glavni dio programskog koda koji izvršava funkciju.

Pomoću Arduino IDE korisnici mogu pisati kod u programsku jezik C/C++ koja je osnova za korištenje Arduina. IDE sadrži mnoge ugrađene funkcije i biblioteke koje olakšavaju rad tako što pružaju gotove funkcije i klase koje se mogu koristiti za razne

zadatke, kao što su upravljanje senzorima, komunikacija s različitim uređajima ili izvođenje specifičnih operacija.

Biblioteke se uključuju u Arduino projekt koristeći naredbu `#include` na početku programa. Ova naredba označava Arduino IDE-u da uključi kod iz određene biblioteke pri kompilaciji. Postoje dvije vrste biblioteka: standardne (ugrađene u Arduino IDE) i korisničke (kreirane od strane programera). Standardne biblioteke obično prate osnovne funkcionalnosti kao što su komunikacija preko serijskog porta, upravljanje analognim i digitalnim ulazima/izlazima itd. Korisničke biblioteke se stvaraju kada programer želi podijeliti svoj kod s drugima ili organizirati složene funkcionalnosti unutar projekta.

Primjer pozivanja biblioteke za RFID senzor koja se piše prije naredbe `void setup`:

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

Slika 1 Primjer pozivanja biblioteke

Izvor: Autor

Također, u programu se nalazi ugrađeni tekstualni uređivač s podrškom za sintaksnu provjeru, označavanje i automatsko dovršavanje koda. Nakon što je kod napisan pomoću komande `verify` provjerava se ispravnost koda, a ukoliko dođe do greške program ju označi. Naredbom `upload` se kod prenosi na programsku pločicu tek nakon što je kod ispravno napisan.

Budući da Arduino koristi jezik C/C++, korisnici mogu koristiti i druge popularne razvojne okoline kao što su Visual Studio Code, Eclipse, Atom i mnoge druge. Ove okoline pružaju dodatne značajke, kao što su bolja podrška za kodiranje, debugiranje i integracija s drugim alatima.

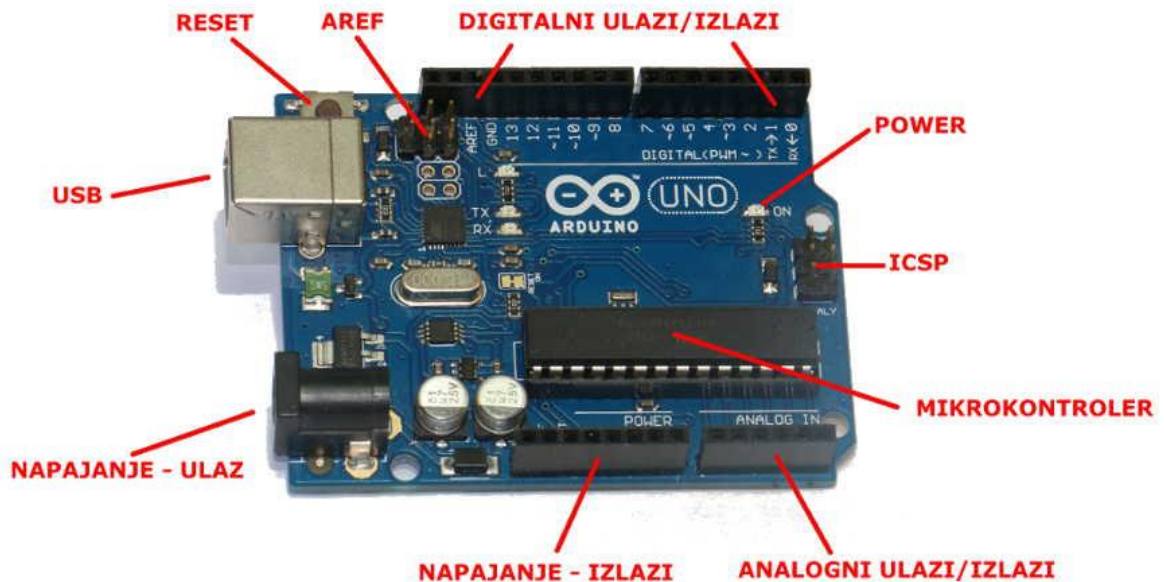
2.4 FUNKCIJE

Arduino programiranje ima u sebi već definirane funkcije koje nije potrebno samostalno izrađivati i koje pomažu pri pisanju i izvršavanju koda. Neke od funkcija su:

- `digitalWrite()` - Postavlja digitalni izlaz na LOW ili HIGH
- `digitalRead()` - Očitavanje vrijednosti digitalnog ulaza
- `pinMode()` - Postavljanje pin mod na INPUT ili OUTPUT
- `delay()` - Vremenska odgoda
- `analogWrite()` - Upisivanje analognog napona na analogni izlaz
- `analogRead()` - Očitavanje analognog napona sa analognog ulaza
- `Serial.print()` - Ispisivanje podataka na serial monitoru

3. DIJELOVI ARDUINA

Arduino Uno je jedna od najpopularnijih i najčešće korištenih Arduino ploča. Sastoji se od nekoliko ključnih dijelova koji omogućuju njezinu funkcionalnost i fleksibilnost.



Slika 2 Opis dijelova Arduino Uno

Izvor: sys.portal CARNET: Arduino i nadzor parametara prostorije (<https://sysportal.carnet.hr/node/1418>)

Pinovi na Arduinou služe radi povezivanja i vršenja komunikacije s drugim vanjskim uređajima koji mogu biti različiti senzori. Arduino sadrži 4 skupine pinova koji su:

- Digitalni pinovi
- Analogni pinovi
- Pinovi napajanje
- ICSP pinovi

3.1 DIGITALNI ULAZI/IZLAZI

Digitalni pinovi na Arduino platformi predstavljaju sučelje koje omogućava povezivanje mikrokontrolera s različitim elektroničkim komponentama, kao što su LED diode, senzori, motorni pogoni i drugi uređaji. Svaki digitalni pin ima svoje karakteristike i može se programirati da izvršava različite zadatke. Također pinovi sadrže ugrađeni otpornik kojim se može pomoću naredbe `digitalWrite` (LOW/HIGH) paliti ili gasiti. Pin 0 (RX) služi za slanje podataka a 1 (TX) za primanje podataka.

Digitalni pinovi rade na radnoj voltaži od 5V za većinu Arduino modela ali postoje i modeli koji rade na 3.3V. Digitalni pinovi rade s digitalnim signalima koji imaju vrijednost 0 ili 1. Programiranje ovih pinova se vrši pomoću naredba `digitalRead()` i `digitalWrite()`.

3.2 ANALOGNI ULAZI/IZLAZI

Analogni pinovi na Arduino platformi su ključna komponenta koja omogućuje uređaju da čita i interpretira analogne signale iz okoline. Za razliku od digitalnih pinova koji rade s digitalnim signalima (visokim i niskim), analogni pinovi omogućavaju mikrokontroleru da mjeri različite razine napona i pretvara ih u digitalne vrijednosti koje možemo koristiti u programiranju.

Analogni pinovi označeni su na Arduino A0, A1, A2, A3, A4, A5. Omogućuju čitanje analognih vrijednosti različitih senzora, kao što su senzori temperature, svjetline, pritiska ili potencijometri. Ako je potrebno izmjeriti neku vrijednost koja se ne može mjeriti sa 0 ili 1 onda koristimo analogne pinove koji mogu mijenjati vrijednost. Arduino mikrokontroleri imaju ugrađeni ADC koji pretvara analogne signale u digitalne vrijednosti. Očitavanja se često skaliraju u vrijednosti između 0 i 1023 (0 V do radnog napona), što omogućava lako razumijevanje i obradu rezultata.

3.3 NAPAЈANJE

Arduino se moԑe napajati pomoću USB kabela ili eksternog izvora napajanja. Arduino ima ugrađeni regulator napajanja koji osigurava stabilno napajanje mikrokontrolera i ostalih uređaja koji su spojeni. Pinovi za napajanje se nalaze lijevo od analognih pinova i označeni su oznakama: 3.3 V, 5 V, GND i Vin, koji su:

- Izlaz 3.3 V osigurava konstanti istosmjerni izvor napajanja od 3.3 V pri struji od 40mA.
- Izlaz 5 V osigurava konstanti istosmjerni izvor napajanja od 5 V pri struji od 40 mA.
- GND ili uzemljenje služi za zatvaranje strujnog kruga
- Vin pin je ulazni napon Arduina kad se koristi vanjsko napajanje

3.4 ICSP ULAZI/IZLAZI

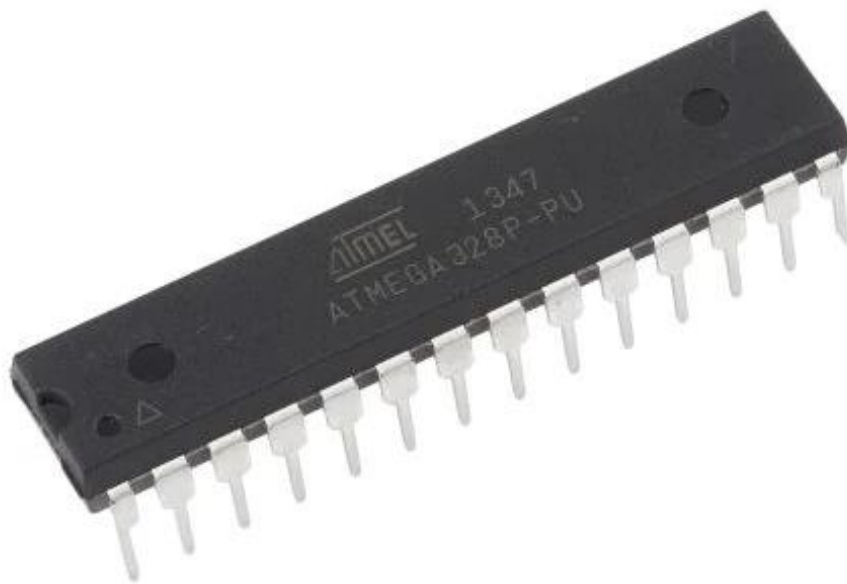
Uz digitalne i analogne pinove napajanja na Arduinou se nalaze i ICSP pinovi (engl. In-circuit Serial Programming) koji služe za ICPS komunikaciju. Sastoje se od dvije skupine po 6 pinova s kojim se moԑe programirati Arduino bez njegovog isključivanja iz strujnog kruga.

3.5 ATmega328

ATmega328 je glavni procesor na Arduino Unu i drugim sličnim modelima. Jedna od ključnih karakteristika ATmega328 mikrokontrolera je njegova 8-bitna arhitektura što znači da da procesor obrađuje podatke u osmerobitnim segmentima, čime je pogodan za jednostavne i srednje funkcije. Radi na radnom taktu od 20 MHz što omogućava brzu obradu podataka i izvođenje različitih zadataka. Ovaj mikrokontroler ima 32 KB interne flash memorije za skladištenje programa i 2 KB SRAM memorije za privremeno čuvanje podataka u toku izvođenja programa.

Velika prednost ovog mikrokontrolera je njegova podrška za serijsku komunikaciju uključujući UART, SPI i I2c protokole, čime se omogućuje komunikacija s drugim uređajima kao što su senzori, eksterne memorijske jedinice ili kompjuteri.

Za programiranje koristi se Arduino IDE, te nakon što je program napisan korisnik ga prenosi na mikrokontroler pomoću USB kabela.



Slika 3 Prikaz ATmega328

Izvor: Microchip ATmega328 (<https://www.microchip.com/en-us/product/atmega328>)

3.6 PREOSTALI DJELOVI

Uz prijašnje objašnjenje pinova na Arduino se još nalazi:

- Restart tipka
- Kristalni oscillator
- Naponski regulator
- Priključak za vanjsko napajanje
- LED indikatri
- USB priključak
- Mikrokontroler, ATmega328

Restart tipka služi da bi u bilo kojem trenu obavljanja programa Arduino stao i počeo iznova. Kristalnim oscilatorom osigurava se frekvencija od 16MHz koja se može prilagoditi korisničkoj potrebi i bez kojeg Arduino ne bi mogao raditi. Naponski regulator osigurava stabilni istosmjerni napon koji treba biti sa što manje oscilacija.

4. SENZORI

Senzori su elektronički uređaji pomoću kojih se mjeri fizičke, kemijske ili biološke veličine i pretvaraju ju u signal pogodan za mikrokontrolere ili ostale elektroničke uređaje koji ga mogu obraditi.

Senzori su ključne komponente u mnogim područjima, uključujući industriju, medicinu, transport, robotiku i mnoge druge. Postoje više vrsta senzora i svaki je dimenzioniran za mjerenje određene fizičke veličine. Neki od najčešćih senzora koji se koriste su senzori temperature, svjetlosti, pokreta, udaljenosti, vlage, tlaka itd. Pomoću senzora moguće je razviti razne aplikacije, eksperimenti i prototipi. Senzori se mogu koristiti za praćenje i detekciju raznih parametara, te su ključne komponente u razvoju pametnih sustava.

Sastoje se od nekoliko važnih komponenti:

- Detektorski element - osnovna komponenta senzora koja reagira na specifičnu promjenu. Ovisno o vrsti senzora element može reagirati na svjetlo, toplinu, zvuk, tlak, vlagu i mnoge druge promjene.
- Transdudktor- komponenta koja je odgovorna za pretvaranje fizičkih promjena u mjerljivu električnu veličinu poput napona, struje ili otpora.
- Elektronička komponenta- služi kao pretvarač signala ili pojačalo te se koristi za obradu električnog signala iz transdukcijskog procesa, te poboljšava osjetljivost, preciznost i očitavanje senzora.

Prema načinu rada senzori se dijele na:

- Kontaktne – moraju biti u kontaktu s objektom mjerenja
- Beskontaktne – nisu u kontaktu s objektom mjerenja

Prema vrsti izlaznog signala:

- Analogni izlaz bez pojačala
- Analogni izlaz s pojačalom
- Pulsni izlaz – ne kontinuiran izlaz, postoje oscilacije
- Binarni signal . ima dva stanja 0 i 1

4.1 RFID SENZOR RC522

RFID senzor RC522 je popularan modul koji se koristi kao čitanje i pisanje RFID tagova. Koristi radio valove za bežičnu komunikaciju s RFID tagovima. Sastoji se od antene koja služi za slanje i primanje radio valova te omogućuju senzoru da komunicira sa RFID tagovima koji su u blizini. Čitač omogućuje senzoru da čita podatke s RFID taga ili da piše podatke na njega. Također RC522 sadrži i mikrokontroler koji upravlja komunikacijom sa tagova i procesiranjem podataka.



Slika 4 Prikaz RFID RC-522 senzor

Izvor: Electronics-lab (<https://www.electronics-lab.com/project/rfid-rc522-arduino-uno-oled-display/>)

Komunikacija između RC522 senzora i RFID tagova obavlja se preko SPI (Serial Peripheral Interface) sučelja. Kada se RFID tag približi senzoru antena šalje radio valove koji se reflektiraju na RFID tagu. Tag zatim prima te valove i šalje podatke natrag senzoru, a senzor čita te podatke i prosljeđuje ih mikrokontroleru na daljnju obradu. RC522 senzor podržava različite vrste RFID tagova, uključujući one koji koriste frekvenciju od 13.56 MHz.

Sastoji se od pinova:

- SDA – (Serial data line) koristi se za serijsku komunikaciju između senzora i mikrokontrolera, te ovaj pin šalje i prima podatke putem SPI (Serial peripheral interface) protokola
- SCK – (Serial clock line) pin je povezan s taktnim signalom SPI komunikacije te kontrolira brzinu prijenosa podataka između senzora i mikrokontrolera
- MOSI – (Master out slave in) pin se koristi za slanje podataka od mikrokontrolera prema senzoru tijekom SPI komunikacije
- MISO – (Master in slave out) pin služi za primanje podataka od senzora prema mikrokontroleru tijekom SPI komunikacije
- RST – (Reset) pin omogućava resetiranje senzora te postavljanje ovog pina na visoku razinu rezultira ponovnim pokretanjem senzora
- NSS/SS – (Slave select) pin se koristi za označavanje određenog senzora ako postoji više uređaja na istoj SPI komunikacijskoj liniji
- 3.3V i GND – (ground) pin 3.3V služi za napajanje dok GND pin služi za zajedničko uzemljenje između senzora i mikrokontrolera
- IRQ – (Interrupt request) pin koji omogućava senzoru da signalizira mikrokontroleru kada se dogodi neki događaj, kao što je očitavanje kartice.

Ovaj senzor također podržava osnovne sigurnosne funkcije kao što su enkripcija i zaštita od neželjenih pristupa, te je dizajniran za nisku potrošnju energije što ga čini pogodnim za uređaje koji se napajaju iz baterijskog izvora.

Za programiranje i upravljanje RFID-RC522 senzorom, korisnici obično koriste različite biblioteke i primjere dostupne za platforme kao što su Arduino i Raspberry Pi. Ove biblioteke olakšavaju proces čitanja i pisanja podataka s RFID tagova, omogućavajući korisnicima da brzo integriraju RFID funkcionalnost u svoje projekte.

4.2 IMPLEMENTACIJA RFID SENZORA U RASKRIŽJE

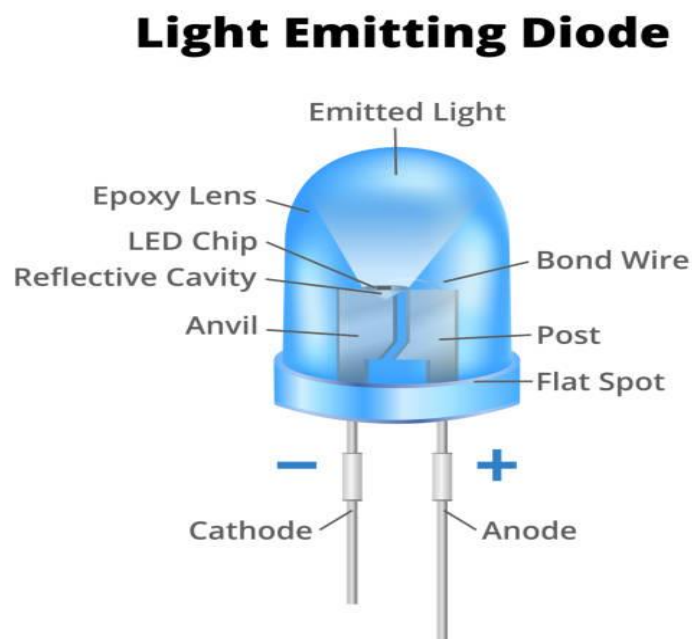
Implementacija RFID (Radio-Frequency Identification) senzora u raskrižje pomoću Arduino platforme može znatno poboljšati upravljanje prometom i omogućiti razne funkcionalnosti, uključujući praćenje vozila, identifikaciju hitnih vozila i poboljšanje sigurnosti. RFID je tehnologija koja omogućava bežično prepoznavanje i praćenje objekata putem radiofrekvencijskih signala. Sastoji se od RFID čitača i RFID tagova. Čitač šalje radiofrekvencijski signal, a tagovi, koji sadrže jedinstveni identifikacijski broj, odgovaraju signalom. Na taj način, čitač može prepoznati tag i izvući informacije iz njega.

Pomoću ove tehnologije svako vozilo koje želi koristiti raskrižje s RFID tehnologijom mora biti opremljeno RFID tagom. RFID tag se može postaviti na vozilo ili na registracijske pločice te sadrži jedinstveni identifikacijski broj. Kada vozilo s RFID tagom pristupi raskrižju, RFID čitač šalje radio frekvencijski signal. RFID tag na vozilu odgovara na taj signal s jedinstvenim identifikacijskim brojem. Arduino mikrokontroler povezan s RFID čitačem interpretira ovu informaciju i identificira vozilo.

Kada se hitno vozilo približi raskrižju s aktivnim RFID tagom, Arduino mikrokontroler može automatski prepoznati vozilo i dati mu prioritet. To se može postići promjenom svjetlosnih semafora ili kontrolom prometnih barijera kako bi se omogućio nesmetan prolaz hitnim vozilima. Arduino mikrokontroler može također spremirati podatke o prolasku vozila kroz raskrižje, uključujući vrijeme i identifikacijski broj vozila. Ovi podaci mogu se koristiti za analizu prometnog toka i poboljšanje sigurnosti na raskrižju. Arduino mikrokontroler može biti povezan s računalom ili pametnim sustavom za daljinsko upravljanje i nadzor raskrižja. To omogućava praćenje prometa u stvarnom vremenu i prilagodbu prometne signalizacije prema potrebama.

5. SVJETLEĆE DIODE

LED dioda je posebna vrsta diode koja svijetli kada je propusno polarizirana. LED diode se često koriste za različite svrhe, uključujući kućno osvjetljenje, dekoraciju, signalizaciju i zaslone. Jedna od najvažnijih karakteristika LED dioda je njihova energetska učinkovitost. U usporedbi s tradicionalnim žaruljama, LED diode troše značajno manje električne energije kako bi proizvele istu količinu svjetlosti. To ih čini ekološki prihvatljivim izborom jer smanjuju potrošnju energije i emisiju stakleničkih plinova. Raznolikost boja je još jedna izvanredna karakteristika LED dioda. Budući da svjetlost LED-ova dolazi iz materijala s različitim svojstvima, moguće je proizvesti širok spektar boja bez upotrebe dodatnih filtera.



Slika 5 Prikaz LED diode

Izvor: iStock (<https://www.istockphoto.com/vector/light-emitting-diode-or-led-diode-with-individual-parts-and-structure-anvil-and-gm1438133992-478727038>)

LED diode se sastoje od nekoliko ključnih komponenti koje omogućuju pretvaranje električne energije u svjetlost koje su:

- LED čip (dioda koja emitira svjetlost): Ovo je srce LED dioda. LED čip je poluvodički element koji emitira svjetlost kada mu prolazi električna struja. Ovisno o materijalu koji se koristi, LED čipovi mogu proizvoditi različite boje svjetlosti.

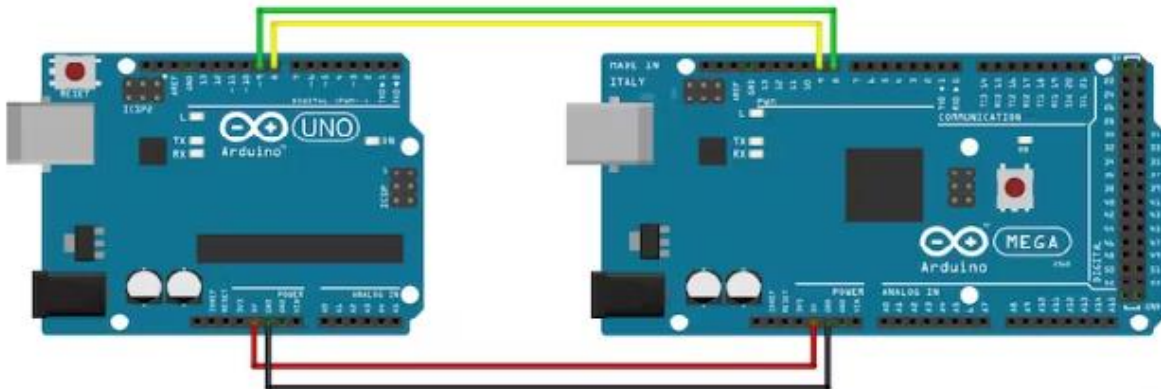
- Reflektor ili leća: Reflektor ili leća nalazi se iznad LED čipa i pomaže usmjeriti svjetlost u željenom smjeru. Ova komponenta može utjecati na raspodjelu svjetlosnog snopa i koncentraciju svjetlosti.

- Kontakti anoda i katoda preko kojih cirkulira električna energija.

6. KOMUNIKACIJA IZMEĐU 2 ARDUINA

Komunikacija između dva Arduino uređaja se odvija putem različitih metoda i protokola. Glavni cilj ovakve komunikacije je omogućiti razmjenu podataka između uređaja kako bi se ostvarila sinergija i koordinacija u radu. Postoje nekoliko načina kako se to može postići:

- Serijska komunikacija (UART): Jednostavan i čest način komunikacije između Arduino uređaja je putem UART (Universal Asynchronous Receiver/Transmitter) komunikacije. Obično Arduino ima ugrađene serijske portove (TX i RX), koji omogućuju razmjenu podataka kroz jednostavne naredbe poput `Serial.print()` i `Serial.read()`. Prilikom korištenja UART-a, potrebno je osigurati da oba uređaja koriste istu brzinu prijenosa (baud rate).



Slika 6 Prikaz serijske komunikacije između dva Arduino

Izvor: Iotguider (<https://iot-guider.com/arduino/serial-communication-between-two-arduino-boards/>)

- I2C komunikacija: I2C (Inter-Integrated Circuit) je dvosmjerni serijski komunikacijski protokol koji omogućuje više uređaja da komuniciraju preko istih vodiča (SDA - Serial Data, SCL - Serial Clock). Arduino može djelovati kao master (glavni) ili slave (podređeni) uređaj. Korištenjem I2C-a, moguće je povezati više uređaja, koristeći samo dva pina.

- SPI komunikacija: SPI (Serial Peripheral Interface) je još jedan serijski protokol koji se često koristi za komunikaciju između Arduino uređaja. Koristi četiri linije za komunikaciju - SCK (Serial Clock), MISO (Master In Slave Out), MOSI (Master Out Slave In) i SS (Slave Select). SPI omogućuje bržu komunikaciju od I2C-a, ali zahtijeva više pina.
- Bežična komunikacija: Arduino također podržava bežične načine komunikacije kao što su Bluetooth, Wi-Fi i RF (Radio Frequency) moduli. Upotrebom bežične komunikacije, dva Arduino uređaja mogu komunicirati bez fizičke veze. Ovo je posebno korisno kada je potrebno ostvariti komunikaciju na većim udaljenostima.
- Radio frekvencijska (RF) komunikacija: RF moduli omogućuju dvosmjernu bežičnu komunikaciju između Arduino uređaja. Ovi moduli koriste različite frekvencije kako bi omogućili pouzdanu komunikaciju na većim udaljenostima.
- Bluetooth komunikacija: Koristeći Bluetooth module, Arduino uređaji mogu međusobno komunicirati putem Bluetooth protokola. Ovo je popularan način za povezivanje uređaja s pametnim telefonima ili računalima.
- Wi-Fi komunikacija: Wi-Fi modul omogućuje dva Arduino uređaja da se povežu putem lokalne bežične mreže. Ova tehnologija omogućuje visoke brzine prijenosa i širok raspon aplikacija.

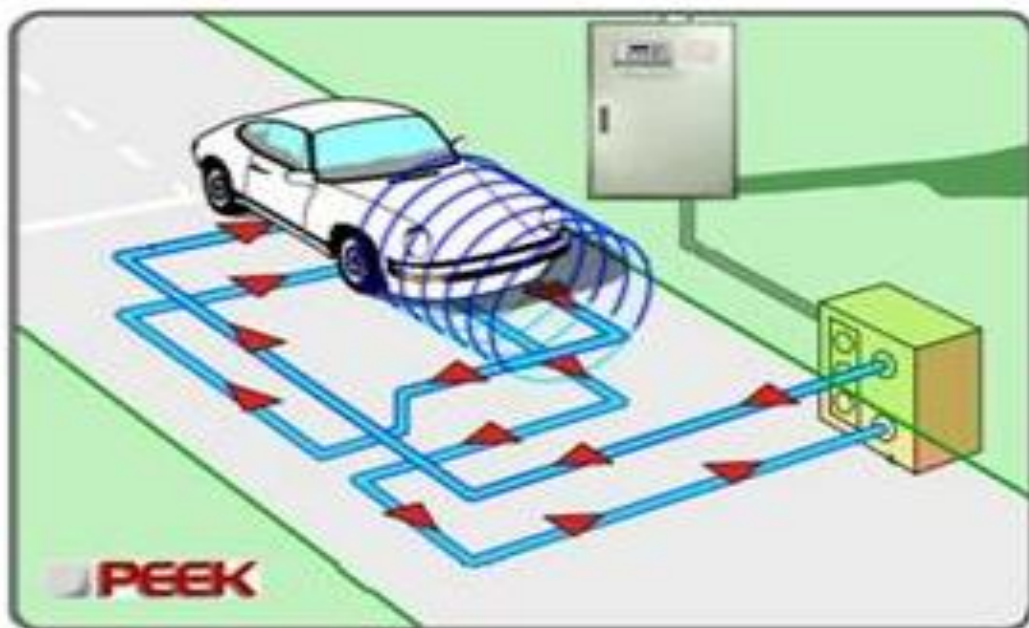
7. PAMETNO RASKRIŽJE

Današnja pametna raskrižja igraju ključnu ulogu u upravljanju prometom i osiguravanju sigurnosti na cestama. Ova raskrižja koriste napredne tehnologije i senzore kako bi optimizirala protok prometa, smanjila gužve, te povećala sigurnost i učinkovitost prometnih tokova. Da bi raskrižje bilo „pametno“, ono mora sadržavati mogućnost detekcije vozila, komunikaciju između senzora i sustava, adaptivnu regulaciju semafora, prioritet za vozila hitne službe, sinkronizaciju raskrižja i pametnu signalizaciju.

Detekcija vozila predstavlja ključan aspekt modernog prometnog sustava omogućujući precizno praćenje kretanja vozila na cesti. Postoji nekoliko različitih metoda detekcija vozila koje se koriste.

Magnetni senzori su često postavljeni ispod površine cesti i reagiraju na prisutnost metalnih predmeta poput vozila. Kada vozilo pređe preko senzora mijenja se magnetsko polje i senzor reagira na promjene. Ova metoda je jednostavna i pouzdana ali može biti osjetljiva na varijacije u težini i materiji vozila.

Također se koriste induksijske petlje koje su ukopane u cestu te kada vozilo prijeđe preko petlje mijenja se inducirani električni otpor i registrira se detekcija. Ova metoda je često korištena na semaforima i parkiralištima.



Slika 7 Prikaz induksijske petlje

Izvor: peek.hr (http://peek.hr/old/induktivne_petlje.htm)

Uz navedene metode također se koriste kamere, radari i akustični senzori. Kamere služe za optičko prepoznavanje vozila analizirajući slike ili video snimke, te sustav može detektirati i pratiti kretanje vozila, ali mogu biti osjetljive na promjene svjetlosnih uvjeta. Radari koriste elektromagnetske valove za mjerenje brzine i udaljenosti vozila. Aktivni radari šalju valove prema vozilima i mjere reflektirane signale, dok pasivni radari detektiraju promjene u elektromagnetskom polju uzrokovanom kretanjem vozila.

Detekcija vozila ima mnoge primjene, uključujući prilagodbu semafora prema prometnom opterećenju, analizu prometnih tokova za urbanističko planiranje, prikupljanje podataka za pametne vozne sustave i mnoge druge. Kombinacija više metoda detekcije može osigurati točne i pouzdane informacije o prometu, što doprinosi boljem upravljanju prometom i stvaranju sigurnijih i učinkovitijih prometnih sustava.

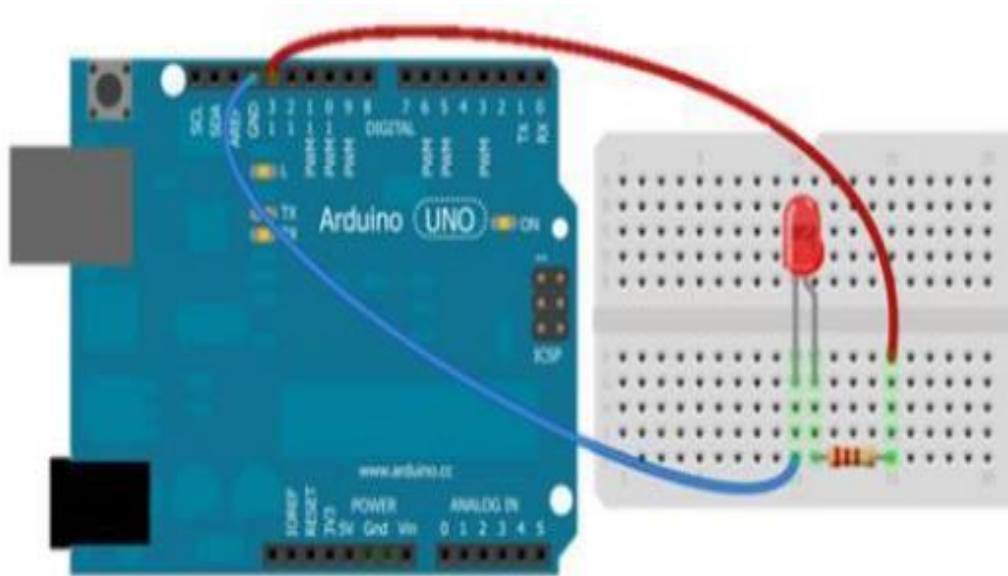
Komunikacija između senzora i sustava na raskrižjima ima ključnu ulogu u optimizaciji prometnog toka, sigurnosti i efikasnosti prometa. Pametna raskrižja koriste napredne tehnologije kako bi senzori detektirali prisutnost vozila i drugih prometnih uvjeta te prenijeli ove informacije centralnom sustavu za upravljanje prometom. Ova komunikacija omogućuje donošenje brzih i informiranih odluka kako bi se osigurala bolja regulacija semafora i smanjenje gužvi. Korištenje ovih tehnologija postavlja temelje za pametne gradove i modernizirane prometne sustave.

Danas se još koristi adaptivna regulacija semafora koja umjesto statičke raspodjele vremena za svaki semafor i pametna raskrižja koriste algoritme kako bi dinamički prilagodili svjetlosne signale prema trenutnom prometnom opterećenju. Na temelju prikupljenih podataka sustav može produljiti zeleno svjetlo na glavnoj cesti ili može skratiti vrijeme trajanja. Također pametna raskrižja mogu pružiti prioritet za žurna vozila i omogućiti im što brži prolazak kroz raskrižje.

8. IZRADA RADA

Prije početka izrade rada potrebno je provjeriti funkcionalnost i ispravnost komponenti. Komponente korištene u ovom radu su LED diode, RFID senzor, baterija od 9V I eksperimentalna pločica te Arduino Uno i Arduino Mega.

LED diode se testiraju tako da se izvor napajanja stavi u eksperimentalnu pločicu i anoda led diode spoje na pozitivni izvor napajanja, a katoda na negativni pol. Svaka led dioda se mora posebno ispitati i ako radi, ona će zasvijetliti. Kako ne bi došlo do izgaranja led diode potrebno je koristiti otpornik koji ograničuje struju koja prolazi kroz nju.



Slika 8 Prikaz testiranja LED diode

Izvor: MaxPhi (<https://www.maxphi.com/led-interfacing-arduino-tutorial>)

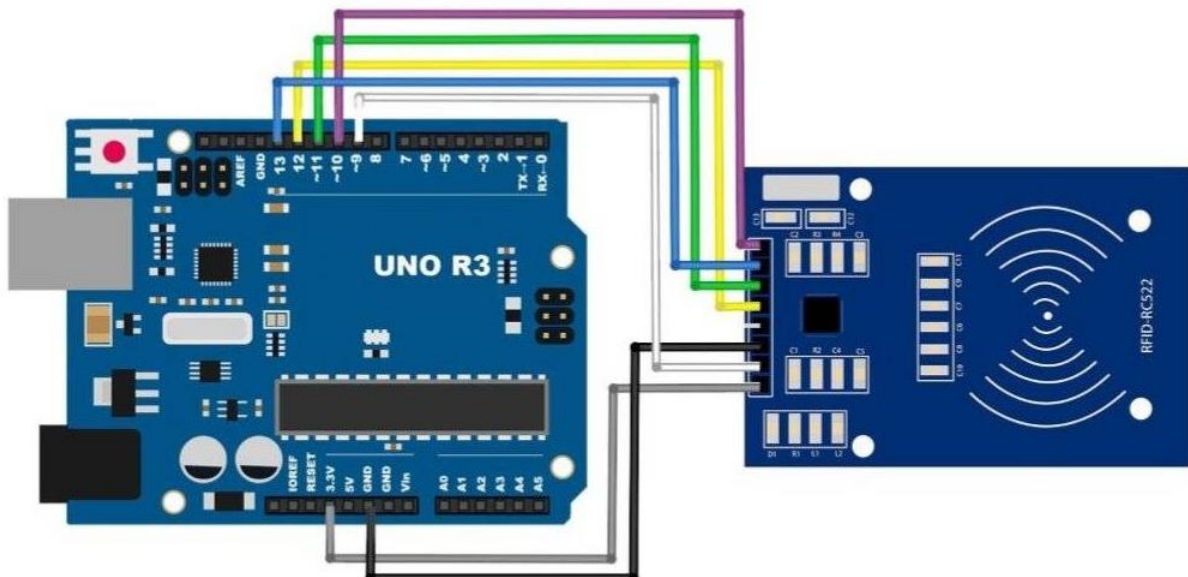
Primjer koda za testiranje led dioda:

```
1
2  int LED = 13;
3
4  void setup() {
5
6    pinMode ( LED, OUTPUT);
7
8  }
9
10 void loop() {
11
12   digitalWrite (LED, HIGH);
13
14 }
15
```

Slika 9 Prikaz koda testiranja LED dioda

Izvor: Autor

Nakon što su led diode testirane potrebno je svaki zasebno RFID senzor testirati je li ispravan. Slika prikazuje kako se izvodi testiranje RFID senzora.



Slika 10 Prikaz testiranja RFID senzora

Izvor: circuits4you.com (<https://circuits4you.com/2018/10/03/interfacing-of-rfid-rc522-with-arduino-uno/>)

Svaki RFID senzor se sastoji od sedam pinova koji su :

- SDA
- SCK
- MOSI
- MISO
- GND
- RST
- 3.3V

Prilikom spajanja RFID senzora u Arduino potrebno je paziti da je svaki pin spojen u odgovarajuće mjesto. Slikom je prikazano na koje pinove se spaja senzor.

Arduino	RFID-RC522
SDA	10
SCK	13
MOSI	11
MISO	12
GND	GND
RST	9
3.3V	3.3V

Slika 11 prikaz spajanja RFID protokola na odgovarajuće Arduino pinove

Izvor: miliohm.com (<https://miliohm.com/mfrc522-rfid-reader-with-arduino-tutorial-the-simplest-way-to-read-rfid-tag/>)

Primjer koda pomoću kojeg se provjerava ako je RFID senzor ispravan:

```
1  #include <SPI.h>
2  #include <MFRC522.h>
3
4  #define SS_PIN1 10
5  #define RST_PIN1 9
6
7  MFRC522 mfrc522_1(SS_PIN1, RST_PIN1);
8
9  void setup() {
10
11  Serial.begin(9600);
12  SPI.begin();
13  mfrc522_1.PCD_Init();
14  Serial.println("RFID spreman");
15  }
16
17  void loop() {
18  if (mfrc522_1.PICC_IsNewCardPresent() && mfrc522_1.PICC_ReadCardSerial()) {
19
20  Serial.write("1"); // kada RFID očita tad printa se 1
21  Serial.println();
22  mfrc522_1.PICC_HaltA();
23  mfrc522_1.PCD_StopCrypto1();
24  }
25  }
26
```

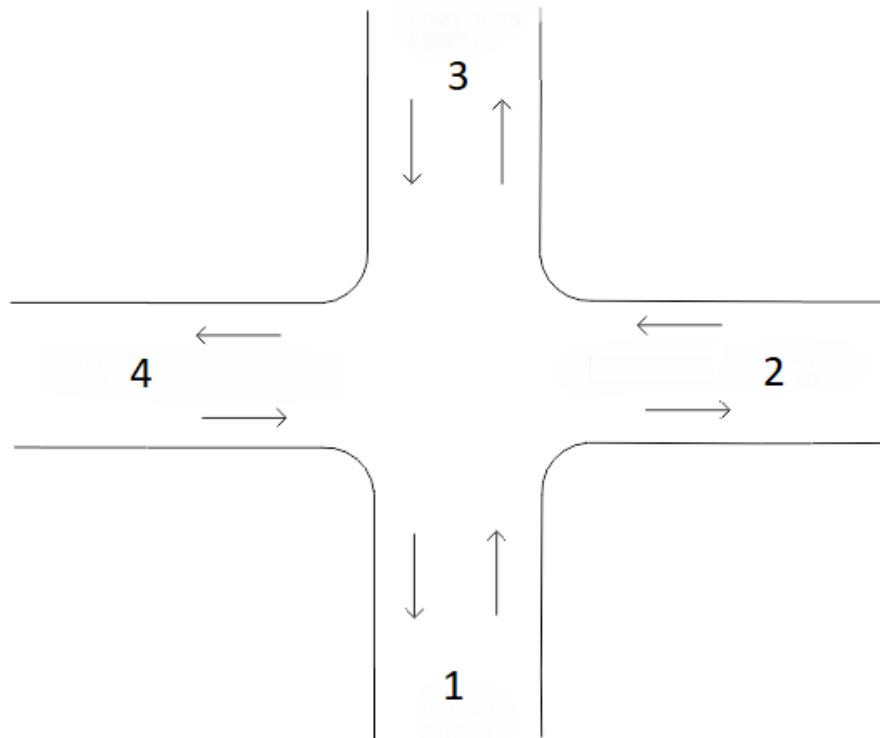
Slika 12 Prikaz koda za testiranje RFID senzora

Izvor: Autor

Zatim kada su sve komponente testirane i ispunjavaju uvjete za rad potrebno je četiri RFID senzora spojiti da rade. Prilikom spajanja do problema je dolazilo kada bi senzori bili blizu jedan drugog. Drugi problem je što kada se četiri RFID senzora spoje na Arduino Uno nema slobodnih pinova za LED diode koje služe kao svijetla od semafora. Zbog tog problema potrebno je koristiti dva Arduino uređaja od kojih je jedan Master a drugi Slave.

Master Arduino u ovom radu je Arduino Uno na kojeg su priključeni četiri RFID senzora i koji pomoću serijske komunikacije šalje informacije Slave uređaju koji je u ovom slučaju Arduino Mega. Na Arduino Mega su spojene sve LED diode koje su semafori raskrižja i one rade kao obično raskrižje sve dok RFID senzor koji je spojen u Arduino Uno ne očita tag i pošalje informaciju na Arduino Megu koji obrađuje te podatke i radi kako mu je zadano u kodu.

Raskrižje je podijeljeno u četiri djela, u svakom djelu se nalazi po jedan RFID senzor, semafor za vozila i dva semafora za pješake.



Slika 13 Prikaz raskrižja

Izvor: resarchgate.net (https://www.researchgate.net/figure/4-way-traffic-intersection-with-4-phases_fig2_264889551)

8.1 PRIKAZ KODA NA MASTER ARDUINU (ARDUINO UNO)

Slikom 14 definirana su ulazno/izlazno mjesto na Arduino Uno za svaki pojedini RFID senzor. Razlog spajanja svakog SS i RST pina omogućuje očitavanje informacije sa svakog posebnog senzora. Naredba `void setup{}` služi za pokretanje serijskog monitora te inicijalizaciju svakog pojedinog RFID MFRC522 senzora.

```

UNOrfid.ino
1  #include <SPI.h>
2  #include <MFRC522.h>
3
4  #define SS_PIN1 10
5  #define RST_PIN1 9
6  #define SS_PIN2 8
7  #define RST_PIN2 7
8  #define SS_PIN3 6
9  #define RST_PIN3 5
10 #define SS_PIN4 4
11 #define RST_PIN4 3
12
13 MFRC522 mfrc522_1(SS_PIN1, RST_PIN1);
14 MFRC522 mfrc522_2(SS_PIN2, RST_PIN2);
15 MFRC522 mfrc522_3(SS_PIN3, RST_PIN3);
16 MFRC522 mfrc522_4(SS_PIN4, RST_PIN4);
17
18 void setup() {
19     Serial.begin(9600);
20     SPI.begin();
21     mfrc522_1.PCD_Init();
22     mfrc522_2.PCD_Init();
23     mfrc522_3.PCD_Init();
24     mfrc522_4.PCD_Init();
25     Serial.println("RFID readers initialized");
26 }

```

Slika 14 Prikaz koda za deklariranje RFID senzora

Izvor: Autor

U ovom djelu koda prikazuje se svaki pojedini RFID senzor, te se provjerava dolazi li do očitavanja, te na kojem senzoru je došlo do očitavanja. Također u kodu se nalazi potvrдна naredba kojom se ispiše broj senzora na kojem je došlo do očitavanja.


```

28 void loop() {
29
30     if (mfr522_1.PICC_IsNewCardPresent() && mfr522_1.PICC_ReadCardSerial()) {
31
32         Serial.write("1");
33         Serial.println();
34         mfr522_1.PICC_HaltA();
35         mfr522_1.PCD_StopCrypto1();
36
37     }
38     if (mfr522_2.PICC_IsNewCardPresent() && mfr522_2.PICC_ReadCardSerial()) {
39
40         Serial.write("2");
41         Serial.println();
42         mfr522_2.PICC_HaltA();
43         mfr522_2.PCD_StopCrypto1();
44
45     }
46     if (mfr522_3.PICC_IsNewCardPresent() && mfr522_3.PICC_ReadCardSerial()) {
47
48         Serial.write("3");
49         Serial.println();
50         mfr522_3.PICC_HaltA();
51         mfr522_3.PCD_StopCrypto1();
52
53     }
54     if (mfr522_4.PICC_IsNewCardPresent() && mfr522_4.PICC_ReadCardSerial()) {
55
56         Serial.write("4");
57         Serial.println();
58         mfr522_4.PICC_HaltA();
59         mfr522_4.PCD_StopCrypto1();
60
61     }
62 }

```

Slika 15 Prikaz koda kod kojeg se provjerava očitavanje na RFID senzoru

Izvor: Autor

8.2 PRIKAZ KODA NA SLAVE ARDUINU (ARDUINO MEGA)

Slika 16 prikazuje pokretanje serijskog monitora, te potvrđnu naredbu u serijskom monitoru. S obzirom da je Arduino Mega spreman za očitavanje podataka sa Master Arduina (Arduino Uno) također je prikazano stanje svakog pina, koji su u ovom slučaju output.

```
22 void setup() {
23     Serial.begin(9600);
24     Serial.println("Ready to receive data from Arduino Uno");
25     pinMode(red1, OUTPUT);
26     pinMode(yellow1, OUTPUT);
27     pinMode(green1, OUTPUT);
28     pinMode(pc1, OUTPUT);
29     pinMode(pz1, OUTPUT);
30
31     pinMode(red2, OUTPUT);
32     pinMode(yellow2, OUTPUT);
33     pinMode(green2, OUTPUT);
34     pinMode(pc2, OUTPUT);
35     pinMode(pz2, OUTPUT);
36
37     pinMode(red3, OUTPUT);
38     pinMode(yellow3, OUTPUT);
39     pinMode(green3, OUTPUT);
40     pinMode(pc3, OUTPUT);
41     pinMode(pz3, OUTPUT);
42
43     pinMode(red4, OUTPUT);
44     pinMode(yellow4, OUTPUT);
45     pinMode(green4, OUTPUT);
46     pinMode(pc4, OUTPUT);
47     pinMode(pz4, OUTPUT);
48 }
49
```

Slika 16 Prikaz pokretanja serijskog monitora

Izvor: Autor

Slikom 17 prikazano je deklariranje LED dioda prema odgovarajućim pinovima u Arduino Mega mikrokontroleru.

```
1  const int red1 = 13;  
2  const int yellow1 = 12;  
3  const int green1 = 11;  
4  const int pc1 = 22;  
5  const int pz1 = 23;  
6  const int red2 = 10;  
7  const int yellow2 = 9;  
8  const int green2 = 8;  
9  const int pc2 = 24;  
10 const int pz2 = 25;  
11 const int red3 = 7;  
12 const int yellow3 = 6;  
13 const int green3 = 5;  
14 const int pc3 = 26;  
15 const int pz3 = 27;  
16 const int red4 = 4;  
17 const int yellow4 = 3;  
18 const int green4 = 2;  
19 const int pc4 = 28;  
20 const int pz4 = 29;
```

Slika 17 Prikaz deklariranja LED dioda

Izvor: Autor

```

245 void normalFlow() {
246
247     if (Serial.available())
248     {
249         return;
250     }
251     digitalWrite(pc1, LOW);
252     digitalWrite(pz1, HIGH);
253     digitalWrite(pc3, LOW);
254     digitalWrite(pz3, HIGH);
255     digitalWrite(pc2, HIGH);
256     digitalWrite(pz2, LOW);
257     digitalWrite(pc4, HIGH);
258     digitalWrite(pz4, LOW);
259     digitalWrite(red1, HIGH);
260     digitalWrite(red3, HIGH);
261     digitalWrite(green2, HIGH);
262     digitalWrite(green4, HIGH);
263     delay(2000);
264     if (Serial.available())
265     {
266         return;
267     }
268     digitalWrite(pc1, LOW);
269     digitalWrite(pz1, HIGH);
270     digitalWrite(pc3, LOW);
271     digitalWrite(pz3, HIGH);
272     digitalWrite(pc2, HIGH);
273     digitalWrite(pz2, LOW);
274     digitalWrite(pc4, HIGH);
275     digitalWrite(pz4, LOW);
276     delay(3000);
277

```

Slika 18 Prikaz stanja semafora

Izvor: Autor

Slikom 18 prikazuje se normalno stanje semafora. Također se prikazuje i provjerava očitavanje vrijednosti senzora na Master Arduinu. Ako vrijednost nije očitana, kod nastavlja sa izvršavanjem. Ukoliko je vrijednost na Master Arduinu očitana, tada program prelazi u određeno stanje s obzirom na očitane vrijednosti.

```

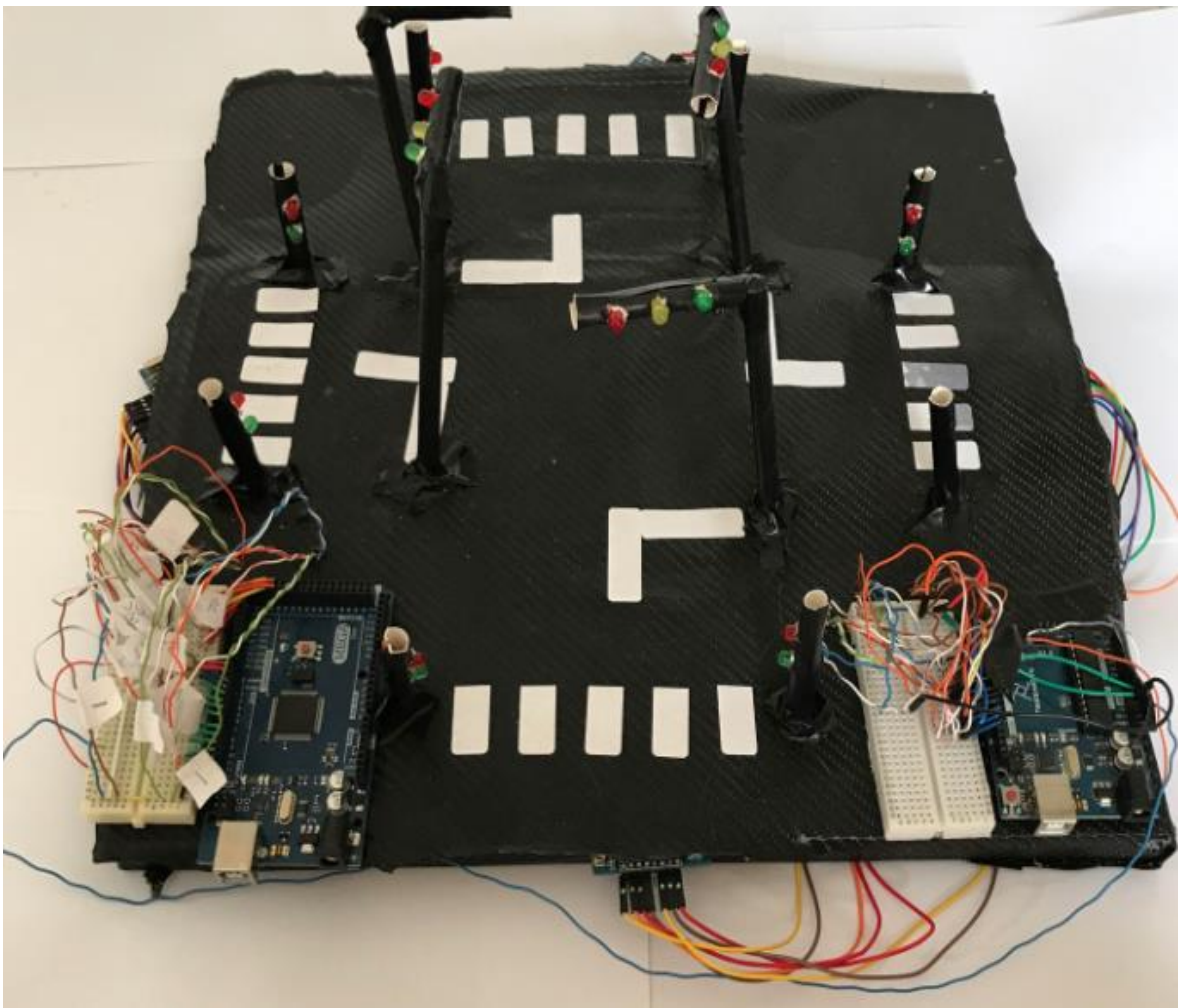
51 void loop(){
52
53     if (Serial.available()) {
54
55         char prefix = Serial.read();
56         Serial.println(prefix);
57         byte uid[4];
58         Serial.readBytes(uid, sizeof(uid));
59
60         if (prefix == '1') {
61             |
62             digitalWrite(red1, LOW);
63             digitalWrite(yellow1, LOW);
64             digitalWrite(green1, HIGH);
65             digitalWrite(red3, HIGH);
66             digitalWrite(yellow3, LOW);
67             digitalWrite(green3, LOW);
68             digitalWrite(red2, HIGH);
69             digitalWrite(yellow2, LOW);
70             digitalWrite(green2, LOW);
71             digitalWrite(red4, HIGH);
72             digitalWrite(yellow4, LOW);
73             digitalWrite(green4, LOW);
74             digitalWrite(pc1, HIGH);
75             digitalWrite(pz1, LOW);
76             digitalWrite(pc2, HIGH);
77             digitalWrite(pz2, LOW);
78             digitalWrite(pc3, HIGH);
79             digitalWrite(pz3, LOW);
80             digitalWrite(pc4, HIGH);
81             digitalWrite(pz4, LOW);
82             delay(5000);
83             digitalWrite(red1, LOW);
84             digitalWrite(yellow1, LOW);
85             digitalWrite(green1, LOW);
86             digitalWrite(red3, LOW);
87             digitalWrite(yellow3, LOW);
88             digitalWrite(green3, LOW);
89             digitalWrite(red2, LOW);
90             digitalWrite(yellow2, LOW);
91             digitalWrite(green2, LOW);
92             digitalWrite(red4, LOW);
93             digitalWrite(yellow4, LOW);
94             digitalWrite(green4, LOW);
95             digitalWrite(pc1, LOW);
96             digitalWrite(pz1, LOW);
97             digitalWrite(pc2, LOW);
98             digitalWrite(pz2, LOW);
99             digitalWrite(pc3, LOW);
100            digitalWrite(pz3, LOW);
101            digitalWrite(pc4, LOW);
102            digitalWrite(pz4, LOW);
103

```

Slika 19 Prikaz stanja nakon očitavanja vrijednosti

Izvor: Autor

Slika 19 prikazuje kod koji se izvršava nakon očitavanja vrijednosti na Master Arduinu. Kodom se očitavanje sa senzora pohranjuje u varijablu char prefix. Nakon toga program provjerava if naredbom o kojem se karakteru radi. Ukoliko dođe do očitavanja na prvom RFID senzoru upalit će se semaforne LED diode zeleno, a sve ostale LED diode (semafor za vozila i pješake) biti će crvene boje sljedećih 5 sekundi. Nakon 5 sekundi ako nije došlo do novog očitavanja, semafor se vraća u normalni tok.



Slika 20 Prikaz izrađenog završnog rada

Izvor: Autor

9. ZAKLJUČAK

U sklopu ovog završnog rada detaljno je istražena implementacija inteligentnog raskrižja koristeći Arduino mikrokontroler s fokusom na pružanje prednosti prolaska hitnim vozilima. Kroz ovaj projekt stečeno je dublje razumijevanje kako današnja tehnologija može unaprijediti i poboljšati upravljanje prometom i osigurati brzu reakciju u hitnim situacijama. Ključni koraci u implementaciji ovog raskrižja uključuju detekciju prisutnosti vozila pomoću senzora, komunikaciju između tih senzora, komunikaciju između hitnih vozila te dinamičku prilagodbu semafora prema prioritetima.

Kroz primjenu Arduino mikrokontrolera i odgovarajućih senzora ostvarena je mogućnost detekcije žurnih vozila te odgovarajućih prebacivanja stanja semafora. Napredak tehnologije omogućuje stvaranje okoline koja poboljšava sigurnost i učinkovitost prometa. Integriranjem prednosti žurnim vozilima u sustavu raskrižja smanjuje se vrijeme reakcije i povećava sigurnost. Kod ovakvih sistema vrlo je bitno osigurati pouzdani rad i smanjiti faktore koji loše utječu na isti.

Unatoč suočavanju s tehničkim izazovima poput senzorskog očitavanja, komunikacijskih protokola i koordinacije svjetlosnih signalizacija, ovaj završni rad stvara temelje za daljnji razvoj i optimizaciju takvih sustava. Kroz inovacije poput ove, može se doprinijeti učinkovitijem upravljanju prometom, smanjenju gužvi i osiguravanju sigurnosti svih sudionika u prometu.

Ovom temom prikazano je kako spajanje tehnologije i inovacije može imati pozitivan utjecaj na našu svakodnevicu, poboljšavajući sustave i procese radi općeg dobra.

LITERATURA

- [1] Arduino mikorkontroler, s Interneta, <https://www.arduino.cc/en/Introduction>, srpanj 2023.
- [2] Smith, A., & Johnson, B. (2018). Design and Implementation of an Arduino-Based Intersection Control System with Emergency Vehicle Priority. *International Journal of Computer Applications*, 179(37), 6-10, kolovoz 2023.
- [3] Serijska komunikacija, <http://www.mcc-us.com/I2CBusTechnicalOverview.pdf>, kolovoz 2023.
- [4] Rahman, S. U., & Rashid, M. S. (2019). Arduino Based Intelligent Traffic Light Control System. *International Journal of Computer Applications*, 182(45), 32-37, kolovoz 2023.
- [5] Adafruit Sensors Biblioteka, s Interneta RFID_MFRC522v2 - Arduino Libraries, kolovoz 2023.
- [6] Banzi, M. (2014). *Arduino Programming in 24 Hours, Sams Teach Yourself*. Sams Publishing, kolovoz 2023.
- [7] Gao, X., Wu, H., & Gao, C. (2018). "Design and Implementation of an Intelligent Street Lighting Control System Using Arduino Uno." In 2018 IEEE 6th International Conference on Smart Energy Grid Engineering (SEGE) (pp. 1-6). IEEE, kolovoz 2023.
- [8] Robinson, D., & Monk, S. (2018). *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education TAB, kolovoz 2023.
- [9] Bauer, M. (2015). *Arduino Electronics Blueprints*. Packt Publishing, kolovoz 2023.
- [10] Nussey, R. (2018). *Arduino for Dummies*. Wiley, kolovoz 2023.

POPIS SLIKA

Slika 1: Primjer pozivanja biblioteke.....	4
Slika 2: Opis dijelova Arduino Uno.....	6
Slika 3: Prikaz ATmega328	9
Slika 4: Prikaz RFID RC-522 senzor.....	11
Slika 5: Prikaz LED diode.....	14
Slika 6: Prikaz serijske komunikacije između dva Arduina.....	16
Slika 7: Prikaz indukcijske petlje	18
Slika 8: Prikaz testiranja LED diode.....	20
Slika 9: Prikaz koda testiranja LED dioda	21
Slika 10: Prikaz testiranja RFID senzora	21
Slika 11: prikaz spajanja RFID protokola na odgovarajuće Arduino pinove.....	22
Slika 12: Prikaz koda za testiranje RFID senzora	23
Slika 13: Prikaz raskrižja.....	24
Slika 14: Prikaz koda za deklariranje RFID senzora	25
Slika 15: Prikaz koda kod kojeg se provjerava očitavanje na RFID senzoru	26
Slika 16: Prikaz pokretanja serijskog monitora.....	27
Slika 17: Prikaz deklariranja LED dioda	28
Slika 18: Prikaz stanja semafora.....	29
Slika 19: Prikaz stanja nakon očitavanja vrijednosti.....	30
Slika 20: Prikaz izrađenog završnog rada	31

KAZALO KRATICA

- Arduino IDE (eng. Integrated Development Enviroment) – arduino programsko okruženje
- GND (eng. Ground) – uzemljenje
- ICSP (eng. In Circuit Serial Programming) – programiranje putem posebnih konektora
- IRQ (eng. Interrupt Request) – mehanizam koji omogućava komponentama da prekinu normalni tijek izvođenja programa
- I²C (eng. Inter-Integradted Circuit) – unutarnji integrirani krug
- LED (eng. Light Emitting Diode) – svjetleća dioda
- MOSI (eng. Master Out Slave In) – pin za serijsku komunikaciju
- MISO (eng. Master In Slave Out) – signalni pin za serijsku komunikaciju
- NSS/SS (eng. Slave Select) – pin za serijsku komunikaciju
- RFID (eng. Radio-Frequency Identification) – identifikacija putem radio-frekvencije
- RF (eng. Radio Frequency) – radio frekvencija
- RST (eng. Reset) – restart
- RX (eng. Receive) – primanje
- TX (eng. Transmit) – slanje
- SCK (eng. Serial Clock) – pin za komunikacijske protokole
- SDA (eng. Serail Dana Line) – komunikacijski protokol
- SPI (eng. Serial Peripheral Interface) – protokol za serijsku komunikaciju
- SRAM (eng. Static Random Access Memory) – statička memorija s nasumičnim pristupom
- UART (eng. Universal Asynchronous Receiver-Transmitter) – komunikacijski modul
- V_{in} (eng. Voltage Input) – napajanje
- Wi-Fi (eng. Wireless Fidelity) – bežično lokalno umrežavanje