

# Mogućnosti primjene Arduino razvojne platforme za simulaciju rada raskrižja

---

Slavulj, Marko

Undergraduate thesis / Završni rad

2018

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:119:367366>

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

*Download date / Datum preuzimanja:* **2025-03-10**



*Repository / Repozitorij:*

[Faculty of Transport and Traffic Sciences -  
Institutional Repository](#)



**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET PROMETNIH ZNANOSTI**

**Marko Slavulj**

**MOGUĆNOSTI PRIMJENE ARDUINO RAZVOJNE PLATFORME ZA  
SIMULACIJU RADA RASKRIŽJA**

**ZAVRŠNI RAD**

**Zagreb, 2018.**

Zagreb, 29. ožujka 2018.

Zavod: **Zavod za inteligentne transportne sustave**  
Predmet: **Računalstvo**

## ZAVRŠNI ZADATAK br. 4752

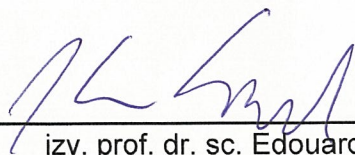
Pristupnik: **Marko Slavulj (0135238122)**  
Studij: **Inteligentni transportni sustavi i logistika**  
Smjer: **Inteligentni transportni sustavi**

Zadatak: **Mogućnosti primjene Arduino razvojne platforme za simulaciju rada raskrižja**

### Opis zadatka:


Sustave upravljanja semaforiziranim raskrižjima u cestovnom prometu moguće je prije implementacije simulirati koristeći razne mikrokontrolere. Jedna od mogućnosti je i razvojna platforma Arduino. U radu je potrebno opisati razvojnu platformu Arduino, napraviti sklopovlje simulatora semafora za jednostavno semaforizirano raskrižje, te napisati potrebni programski kôd.

Mentor:



izv. prof. dr. sc. Edouard Ivanjko

Predsjednik povjerenstva za  
završni ispit:

  
Dino Čakija, mag.ing.el., pred. (sumentor)

**Sveučilište u Zagrebu**

**Fakultet prometnih znanosti**

**ZAVRŠNI RAD**

**MOGUĆNOSTI PRIMJENE ARDUINO RAZVOJNE PLATFORME ZA  
SIMULACIJU RADA RASKRIŽJA**

**POSSIBILITY OF APPLYING THE ARDUINO DEVELOPMENT PLATFORM  
FOR SIMULATION OF SIGNALIZED INTERSECTIONS**

Mentor: izv. prof. dr. sc. Edouard Ivanjko

Student: Marko Slavulj

Sumentor: dipl. ing. Dino Čakija

JMBAG: 0135238122

**Zagreb, kolovoz 2018.**

## SAŽETAK

U ovom završnom radu obrađuje se problematika upravljanja jednostavnim semaforiziranim raskrižjem primjenom adaptivnog upravljanja implementiranog u mikrokontroleru. Iz svakidašnjice znamo da je prometna potražnja sve veća i da fiksna (ustaljena) upravljanja semaforiziranim raskrižjem nemaju mogućnost praćenja te promjene prometne potražnje. Za rješenje ovog problema, istražena je mogućnost primjene mikrokontrolera za adaptivno upravljanje semaforiziranim raskrižjem. Načini na koji će se dokazati efikasnost sustava su uporaba simulacije platforme „Arduino“ na maketi, te matematičkim proračunima. Simulacija jednostavnog adaptivno upravljivog semaforiziranog raskrižja prikazuje mogućnost produljenja ili skraćanja trajanja zelenog svjetla uz fiksno trajanje ciklusa kao i upravljanje najave pješaka koristeći „Arduino“ razvojnu platformu sa svim pripadajućim sklopovljem, kao i napisanim programskim kôdom i dijagramom toka koji upravljaju radom ovim adaptivnim upravljanjem semaforiziranim raskrižjem.

KLJUČNE RIJEČI: Arduino, ciklus, faza, semaforizirano raskrižje, prometni tok.

## SUMMARY

In the final paper, "Possibility of applying the Arduino development platform for simulation of signalized intersections", the issues of managing a simple semaphorized intersection through using adaptive control is processed. The purpose of the work is to connect the acquired knowledge, upgrade the same, and share it with others. The aim of the paper is to demonstrate the use of a microcontroller for adaptive management of the semaphorized intersection. The theories to prove the theory are the use of the "Arduino" platform simulation on the model and the mathematical calculations. The paper describes the simulation of a simple adaptive manageable semaphorized intersection using the "Arduino" development platform with all the accompanying hardware, as well as written programming code and flowcharts that manage the work of an adaptive semaphorized intersection.

KEY WORDS: Arduino, cycle, phase, semaphorized intersection, traffic flow.

# SADRŽAJ

1. UVOD .....	1
2. PROBLEM UPRAVLJANJA SEMAFORIZIRANIM RASKRIŽJEM .....	4
2.1. Obilježja semaforiziranih raskrižja .....	5
2.2. Ulazne veličine za analizu semaforiziranog raskrižja .....	5
2.3. Fiksno upravljanje semaforiziranim raskrižjem .....	6
2.4. Prometno ovisno upravljanje .....	7
2.4.1. Djelomično prometno ovisno upravljanje .....	8
2.4.2. Potpuno prometno ovisno upravljanje .....	8
2.4.3. Upravljanje na osnovi protoka/gustoće (Adaptivno upravljanje) .....	9
2.5. Određivanje signalnog plana .....	10
2.5.1. Ciklus .....	11
2.5.2. Faza .....	12
2.5.3. Zaštitno međuvrijeme .....	14
2.5.4. Efektivno trajanje zelenog svjetla .....	16
2.5.5. Prijelazna vremena .....	17
3. RAZVOJNO OKRUŽENJE ARDUINO .....	18
3.1 Arduino ATmega 2560 – pinovi .....	18
3.2. Arduino osjetila .....	20
4. SKLOP ZA SIMULACIJU SEMAFORIZIRANOG RASKRIŽJA .....	22
4.1. Princip rada semaforiziranog raskrižja .....	23
4.2. Izrada makete semaforiziranog raskrižja .....	26
4.2.1. Izrada dokumentacije .....	27
4.2.2. Moguće nadogradnje .....	28
5. PROGRAM ZA UPRAVLJANJE SEMAFOROM .....	30

6. TESTIRANJE SIMULATORA.....	33
7. ZAKLJUČAK.....	36
LITERATURA.....	38
POPIS SLIKA.....	39
POPIS TABLICA .....	40
POPIS FORMULA .....	41
POPIS KRATICA.....	42
PRILOG 1. PROGRAMSKI KÔD ZA UPRAVLJANJE SEMAFORIZIRANIM RASRIŽJEM .....	43

## 1. UVOD

Svakim danom prometna potražnja je sve više u porastu. Gradovi su zagušeni cestovnim motornim prometom što smanjuje kvalitetu življenja i utječe na zdravlje ljudi te je iz tog razloga potrebno pronaći rješenje problema zagušenja. Rješenje nije proširiti prometnice, već se okrenuti novim tehnologijama, odnosno Inteligentnim Transportnim Sustavima (ITS). ITS je grana koja se bavi nadgradnjom klasičnog prometnog sustava informacijsko-komunikacijskim sustavima sa svrhom bržeg, sigurnijeg, ekološki prihvatljivijeg i učinkovitijeg transporta putnika i roba.

Dio ITS-a čine i razni servisi upravljanja prometom kao što je to adaptivno upravljanje semaforiziranim raskrižjem. Iz kojeg razloga je to potrebno? Kao što je rečeno, prometna zagušenja su sve veća. Naročito u urbanim sredinama. Na raskrižjima gdje se promet odvija fiksnim signalnim planovima, odnosno ustaljenim upravljanjem svjetlosnim signalima (semaforima) dolazi do nepotrebnih zaustavljanja prometnih tokova. Iz tog razloga je potrebno primijeniti adaptivno upravljanje prometnim tokom kako bi se povećala propusna moć raskrižja promjenom signalnih planova raskrižja u odnosu na prometnu potražnju. U odnosu na fiksno upravljanje svjetlosnim signalima, adaptivni sustav je znatno složeniji ali i bitno efikasniji.

U ovom završnom radu bit će prikazan način rada adaptivnih sustava upravljanja semaforiziranim raskrižjem u teoriji i maketi koja je upravljana Arduino razvojnom platformom kao i svim ostalim potrebnim parametrima za realizaciju takvog sustava. Cilj rada je opisati razlog implementacije i korištenja takvih sustava, doprinijeti u stvaranju pozitivne slike na kompletan sustav zato što je grana ITS-a relativno novi pojam u Republici Hrvatskoj, dodati zaključak o mogućnosti primjene mikrokontrolera za izradu maketa semaforiziranih raskrižja na kraju rada te upotpuniti i postojeće znanje o toj temi. Naslov ovog završnog rada je: Mogućnosti primjene Arduino razvojne platforme za simulaciju rada raskrižja. Rad je podijeljen u sedam cjelina:

### 1. Uvod;



2. Problem upravljanja semaforiziranim raskrižjem;
3. Razvojno okruženje Arduino;
4. Sklop za simulaciju semaforiziranog raskrižja;
5. Program za upravljanje semaforima;
6. Testiranje simulatora;
7. Zaključak.

U drugom poglavlju, opisan je problem upravljanja semaforiziranim raskrižjem kao što je to vršno opterećenje, ponašanje sudionika u prometu, nedovoljno precizan podatak brojanja prometa i tako dalje. Na-dalje, opisani su prometni parametri semaforiziranih raskrižja gdje je najvažniji pojam za napomenuti ciklus i faza. Projektiranje semaforiziranih raskrižja ne može započeti bez ulaznih veličina, a semaforizirana raskrižja se mogu upravljati fiksnim, prometno ovisnim i adaptivnim upravljanjem signalnim planovima.

Za realizaciju završnog rada između ostalog potrebna je i razvojna platforma s mikrokontrolerom. Upravo u trećem poglavlju će biti spomena o takvoj platformi koja se zove Arduino. Razmatrat će se o njegovim tehničkim karakteristikama, načinu upotrebe, ulazno/izlaznim pinovima koji služe za komunikaciju s vanjskim komponentama sustava te osjetilima koji se koriste na izrađenoj maketi za simulaciju rada jednostavnog semaforiziranog raskrižja.

Da bi sustav radio, potrebno je načiniti sklop. Taj dio realiziran je u četvrtom poglavlju gdje je opisan sklop sustava za upravljanje pomoću električne sheme i eksperimentalne makete komponenti semaforiziranog raskrižja, izrada makete jednostavnog semaforiziranog raskrižja kroz faze kao i izrada potrebne dokumentacija u Fritzing programskoj podršci te na posljatku i mogućnosti dorade sustava. Princip rada dodatno je opisan, radi razumijevanja ideje rješavanja problematike upravljanja semaforiziranim raskrižjem.

U petom poglavlju, proći će se kroz programski kôd i dijagram toka. Time će biti jasna ideja upravljanja semaforiziranim raskrižjem uz sve prometne zahtjeve.

Testiranje simulatora, opisat će se u šestom poglavlju. Tu će biti prikazan signalni plan te će se po njemu opisati slijed izvršavanja izmjena signalnih grupa uz ulazne parametre najave pješaka i protoka vozila u raskrižje.

Rad završava zaključkom te prijedlozima za nastavak rada na ovoj temi.

## 2. PROBLEM UPRAVLJANJA SEMAFORIZIRANIM RASKRIŽJEM

Čimbenici kao što su ponašanje sudionika u prometu, vremenski uvjeti, vršna opterećenja, prometna infrastruktura te korisnički zahtjevi imaju bitan utjecaj na upravljanje semaforiziranim raskrižjem. Sustavi upravljanja mogu biti fiksni, prometno ovisni i adaptivni. Fiksno upravljanje signalnim planom nema mogućnost izmjene trajanja signalnih grupa s obzirom na stvarnovremenske prometne podatke dobivene iz raznih senzorskih sustava, prometno ovisno upravljanje ima zadaću detekcije vozila na pojedinim privozima semaforiziranog raskrižja i prema tome produljuje ili skraćuje trajanje faza, dok se adaptivno upravljanje temelji na prometnim parametrima protoka i/ili gustoće prometa. Pješaci su najugroženiji sudionici u prometu. Iz tog razloga kod projektiranja raskrižja treba posebno voditi računa o zahtjevima pješaka. U gušćim naseljima te u područjima trgovačkih centara, često su uporabni zahtjevi pješaka važniji od onih koje iziskuje promet na prometnim tokovima semaforiziranog raskrižja. Treba težiti da pješaci prijeđu cijeli kolnik u jednom mahu, čime bi se izbjeglo zadržavanje pješaka na otoku [1].

Kod semaforiziranog raskrižja sve su veće potrebe za prilagodbom upravljanja s obzirom na prirodu odvijanja prometnog sustava. Danas, dok je promet konstantno u porastu, izračunati optimalni signalni planovi na temelju brojanja prometa ne mogu točno predvidjeti prometnu situaciju na raskrižjima. To znači nedovoljno dobro vođenje prometa kroz raskrižje koje naznačuje ekološka zagađenja i prometna zagušenja. Rješenje je moguće pronaći u integraciji sustava upravljanja s drugim ITS rješenjima u svrhu povećanja kvalitete gradskog prometnog sustava. Naime, na mreži raskrižja gdje se promet regulira ustaljenim i nekoordiniranim svjetlosnim signalima dolazi do nepotrebnih zaustavljanja i prekidanja prometnih tokova. ITS rješenje adaptivnog vođenja prometnog toka povećava propusnu moć tako da se redosljed odlučivanja i trajanje ciklusa stalno prilagođavaju promjenjivim potrebama prometnog toka i uvjetima okruženja. U odnosu na koordiniran ustaljeni režim rada semafora, adaptivni sustav je složeniji, ali bitno učinkovitiji jer smanjuje ukupne vremenske gubitke i ostale pokazatelje kvalitete prometnog sustava [2].

## **2.1. Obilježja semaforiziranih raskrižja**

Raskrižje je prometna površina na kojoj se križaju ili spajaju dvije ili više cesta [1]. Semaforizirano raskrižje je vrsta raskrižja upravljiva semaforima (svjetlosnom signalizacijom). Najvažniji pojmovi pri tome su ciklus i faza. Ciklus je trajanje signalnog plana, odnosno vrijeme za koje se izmjene sve faze u tom ciklusu. Faza je dio ciklusa u kojem pojedini prometni tokovi imaju istovremeni slobodan prolaz. Ciklus se dijeli na faze, a svaka faza se sastoji grupa. Svaka grupa sastoji se od zelenog, žutog i crvenog signalnog pojma. Zato je osnovna zadaća semaforizacije određivanje tri vremenske veličine: ciklus, zeleno vrijeme i zaštitno vrijeme [1]. Semaforizirana raskrižja mogu biti fiksno, prometno ovisna i adaptivno upravljiva.

## **2.2. Ulazne veličine za analizu semaforiziranog raskrižja**

Za projektiranje i analizu semaforiziranog raskrižja potrebne su ulazne veličine prikazane tablicom 1. Navedenom tablicom se ukazuje da projektant raskrižja treba voditi računa o geometrijskim i prometnim uvjetima, uvažavajući mogućnost semaforizacije (veća raskrižja s povećanim prometom). Geometrijski uvjeti opisuju tehničke karakteristike raskrižja, kao što su: tip područja u kojem se raskrižje nalazi, broj i širina trakova, uzdužni nagib privoza, prostor za parkiranje u zoni raskrižja itd. Kod prometnih uvjeta opisuje se utjecaj vanjskih elemenata kao što su: faktor vršnog sata, pješački promet, dozvoljena prilazna brzina itd. Na posljatku, semaforizacija opisuje proračune signalnog plana, kao što su: duljina trajanja ciklusa, broj faza, itd.

Tablica 1. Ulazne veličine za analizu semaforiziranog raskrižja [1]

GEOMETRIJSKI UVJETI	<ul style="list-style-type: none"> <li>- Tip područja u kojem se raskrižje nalazi</li> <li>- Broj trakova na privozima</li> <li>- Širina trakova</li> <li>- Uzdužni nagib privoza</li> <li>- Posebni ili zajednički trakovi za lijevo i desno skretanje</li> <li>- Proctor za parkiranje u zoni raskrižja</li> </ul>
PROMETNI UVJETI	<ul style="list-style-type: none"> <li>- Brojanje prometa</li> <li>- Osnovni zasićeni tok</li> <li>- Faktor vršnog sata</li> <li>- Utjecaj teških (teretnih) vozila</li> <li>- Pješački i biciklistički promet</li> <li>- Stajališta javnog prometa u području raskrižja</li> <li>- Broj parkiranja u području raskrižja</li> <li>- Način dolaska vozila na raskrižje</li> <li>- Broj vozila koja dolaze na početku zelene faze</li> <li>- Dozvoljena prilazna brzina</li> </ul>
SEMAFORIZACIJA	<ul style="list-style-type: none"> <li>- Duljina ciklusa</li> <li>- Broj i raspored odvijanja faza</li> <li>- Zeleno svjetlo</li> <li>- Zaštitno vrijeme</li> <li>- Tip upravljanja prometom</li> <li>- Minimalna zelena vremena za pješake</li> </ul>

### 2.3. Fiksno upravljanje semaforiziranim raskrižjem

Signalni plan izračunava se na temelju dobivenih podataka o brojanju prometa (broj vozila, klasifikacija prema vrsti vozila, smjeru kretanja). Signalni plan je fiksnog (ustaljenog) trajanja i nema izmjene trajanja signalnih grupa s obzirom na stvarnovremenske podatke dobivene iz raznih senzorskih sustava. Moguća je promjena fiksnih signalnih planova (u samom semaforskom uređaju) prema dnevnim promjenama prometne potražnje [3].

## 2.4. Prometno ovisno upravljanje

Zasniva se na detekciji vozila na pojedinim privozima semaforiziranog raskrižja. Algoritmi upravljanja imaju konačan broj mogućih rješenja. Za primjer, ukoliko na privozu ima više od 10 vozila u repu čekanja, trajanje zelenog svjetla se produljuje za dodatnih  $n$  sekundi ili ukoliko na privozu nema prisustva vozila, „preskoči“ fazu. Prednosti ovog sustava su:

- Smanjenje kašnjenja na prolazak raskrižjem;
- Prilagodba na kratkoročne varijacije prometnog toka (po privozima);
- Povećanje kapaciteta.

Postoje tri osnovne vrste prometno ovisnog upravljanja s razlikom u načinu korištenja semaforiskog uređaja [3]:

- Djelomično prometno ovisno upravljanje;
- Potpuno prometno ovisno upravljanje;
- Upravljanje na osnovu protoka/gustoće (adaptivno upravljanje).



Slika 1. Prikaz detekcije vozila [3]

Slika 1 prikazuje detekciju vozila s dvije petlje (detektora) postavljenih na razmaku od 60 m na jednom privozu. Prvi detektor postavljen je na razmaku od 40 m od zaustavne linije, dok je

drugi postavljen 100 m od zaustavne linije. Prvo (plavo) vozilo je brojano vozilo koje napušta petlju, drugo vozilo (bijelo) je vozilo koje se nalazi na petlji, a brzina kretanja je omjer jednog kvadratića i pola sekunde.

#### **2.4.1. Djelomično prometno ovisno upravljanje**

Koristi se na koridorima gdje postoji glavni uniformni tok, te relativno malo opterećenje na sporednim privozima. Detektori se postavljaju samo na sporednim privozima. Aktivna je faza glavnog prometnog toka dok se ne detektira prisustvo vozila na sporednim privozima. Nakon poštivanja minimalnog zelenog svjetla ta glavnom prometnom toku, aktivira se faza sporednih privoza [3].

Prednosti:

- Efikasna uporaba u koordiniranom djelu prometne mreže;
- Smanjuje kašnjenje na glavnom toku;
- Potreban je manji broj detektora ( samo na sporednim privozima);
- Glavni tok ima maksimalnu iskorištenost prolaska (trajanja zelenog svjetla) [3].

Mane:

- Održavanje sustava;
- Složenost izrade sustava.

#### **2.4.2. Potpuno prometno ovisno upravljanje**

Koristi se na raskrižjima gdje je udio prometa podjednako raspodijeljen po svim privozima. Detektori su postavljeni na svim privozima. Nema pravila u redosljedu faza, trajanju faze (do maksimalnog trajanja zelenog svjetla) i trajanju ciklusa. Ovakvo upravljanje je efikasno za dvofazna i višefazna raskrižja [3].

Prednosti:

- Smanjenje kašnjenja u odnosu na ustaljeni signalni plan;
- Osjetljivo na trenutno prometno opterećenje;
- Informacija o prisustvu (detekciji) vozila, omogućuje efikasnu raspodjelu ciklusa s izmjenama u svakom sljedećem ciklusu;
- Ukoliko nema prisustva vozila na određenom privozu, omogućeno je "preskakanje" faza. Neiskorišteno vrijeme raspodjeljuje se potrebnim privozima/fazama [3].

Mane:

- Održavanje sustava;
- Složenost izrade sustava.

#### **2.4.3. Upravljanje na osnovi protoka/gustoće (Adaptivno upravljanje)**

To je sustav koji vrši prilagodbu signalnih planova na osnovi:

- Trenutne prometne situacije;
- Prometnih zahtjeva;
- Kapaciteta prometne mreže.

Sustav koristi algoritme koji prilagođavaju:

- Omjer trajanja zelenog signalnog plana prema ciklusu;
- Vremenski pomak signalnog plana;
- Trajanje faza;
- Redoslijed faza.

Postoje razlike vodećih razvijenih sustava kao što su: brzina odziva sustava, pristup razvoju algoritma te načinu detekcije vozila. Prvi razvoji adaptivnog upravljanja bili su 1970-ih godina zbog različitih dnevnih promjena prometa, te nemogućnosti korištenja ustaljenog signalnog plana. Dva najpoznatija i najkorištenija sustava su SCATS i SCOOT [3]. Noviji sustavi su UTOPIA te InFlow.



Razlozi uvođenja adaptivnih sustava:

- Potreba za upravljačkim sustavima koji mogu zadovoljiti potrebu upravljanja pri velikim dnevnim promjenama prometne potražnje;
- Smanjenje troškova reprogramiranja signalnih planova (svakih 3 – 5 godina) zbog konstantnog rasta prometne potražnje;
- Upravljanje konfliktima između cestovnog prometa i ostalih modova;
- Smanjenje ukupnih troškova prometnog sustava;
- Upravljanje prometom u „posebnim“ situacijama [3].

Prednosti adaptivnih sustava upravljanja prometom:

- Poboljšanja prometnih parametara kvalitete prometa;
- Kašnjenje (direktan utjecaj na razinu uslužnosti);
- Povećanje prosječne brzine određenom dionicom;
- Bespotrebno ažuriranje ustaljenih signalnih planova (troškovi).

Nedostaci adaptivnih sustava upravljanja prometom [3]:

- Povećana početna ulaganja za infrastrukturu i implementaciju centra za nadzor i upravljanje prometom (potrebna zamjena zastarjelih semaforskih uređaja, uspostava komunikacije između semaforskih uređaja i glavnog upravljačkog centra);
- Veći troškovi održavanja same infrastrukture;
- Otežan proces implementacije i kalibracije sustava.

## **2.5. Određivanje signalnog plana**

Prema međunarodnom sustavu koji se primjenjuje i u Republici Hrvatskoj, redosljed uključivanja signalnih pojmova (svjetala semafora) je (zeleno-žuto-crveno) i (crveno-crveno i žuto-zeleno). Crveno svjetlo ostaje upaljeno i za vrijeme žutog signalnog pojma prije promjene na zeleno svjetlo. Odnos između vremena trajanja zelenog intervala i ukupne duljine ciklusa posebno je važno za optimalnu propusnu moć [4].

U cilju pronalaženja optimalnog signalnog plana za semaforizirano raskrižje potrebno je odrediti sljedeće osnovne pojmove: ciklus, faza, zaštitno međuvrijeme, efektivno zeleno svjetlo, prijelazno vrijeme.

### 2.5.1. Ciklus

Ciklus je trajanje signalnog plana (Slika 2), odnosno vrijeme za koje se izmjene sve faze u tom ciklusu. Matematički, to je zbroj efektivnih zelenih i izgubljenih vremena [5].

Duljina ciklusa (C) može se izračunati jednadžbama (1) i (2):

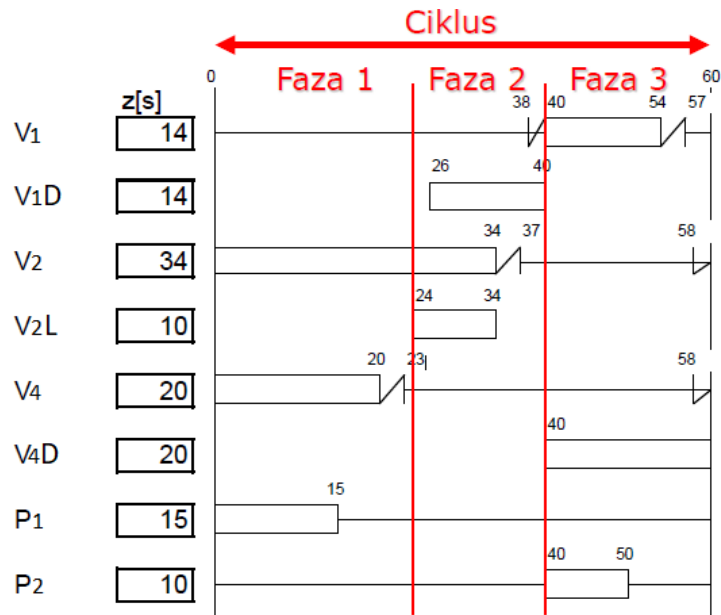
$$C = \sum_{i=1}^n g_i + \sum_{i=1}^n l_i = \sum_{i=1}^n g_i + L \text{ [s]}, \quad (1)$$

$$C = \frac{X_c \cdot L}{X_c - Y} \text{ [s]}, \quad (2)$$

gdje je:

- C – duljina ciklusa [s];
- L – ukupno izgubljeno vrijeme u ciklusu [s];
- $g_i$  - duljina efektivnog zelenog svjetla [s];
- $l_i$  – izgubljeno vrijeme po fazama [s];
- $X_c$  - stupanj zasićenja raskrižja;
- $Y = \sum_{i=0}^n y_i$  - stupanj opterećenja (iskorištenja) raskrižja;

- $\sum_{i=1}^n g_i$  - zbroj efektivnih zelenih vremena vozačkih signalnih grupa;
- $\sum_{i=1}^n l_i$  – zbroj izgubljenih vremena vozačkih signalnih grupa.



Slika 2. Pregled signalnog plana [5]

Duljina ciklusa [5]:

- minimalna (zbroj zaštitnih i minimalnih zelenih vremena);
- maksimalno preporučena: 120 s (> 120 s posebni razlozi);
- optimalna (po nekom kriteriju, funkciji cilja);
- prosječna: za prometno ovisno upravljanje.

### 2.5.2. Faza

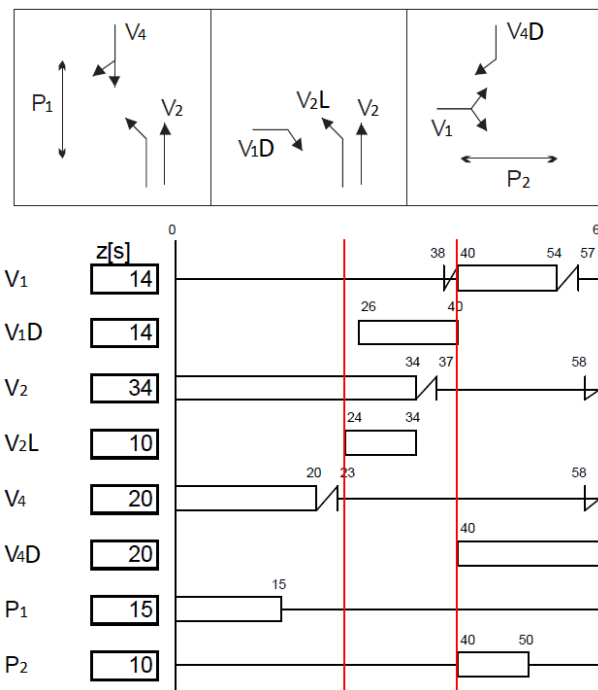
Faza je dio ciklusa u kojem pojedini prometni tokovi imaju istovremeni slobodan prolaz (Slika3). Za upravljanje odvijanjem prometa pomoću semafora (svjetlosnog signalnog uređaja) na nekom raskrižju potrebne su najmanje dvije faze [6].

Duljina trajanja faze ( $f_i$ ) može se izračunati prema izrazu (3):

$$f_i = (C - L) \cdot \frac{y_i}{Y} \text{ [s]}, \quad (3)$$

gdje je:

- $f_i$ - duljina trajanja ciklusa [s];
- $C$ - duljina ciklusa [s];
- $L$  – ukupno izgubljeno vrijeme u ciklusu [s];
- $Y = \sum_{i=0}^n y_i$  - stupanj opterećenja (iskorištenja) raskrižja;
- $y_i$ - stupanj opterećenja (iskorištenja) grupe prometnih traka.



Slika 3. Pregled faza [5]

### 2.5.3. Zaštitno međuvrijeme

Vrijeme je između dvije konfliktne signalne skupine koje slijede uzastopno, tj. vrijeme između faza potrebno za siguran prolazak korisnika raskrižjem (Slika 4).

Preporuka je za motorna vozila (4):

$$t_z \geq t_{\bar{z}} + t_{c\bar{z}} [s], \quad (4)$$

gdje je:

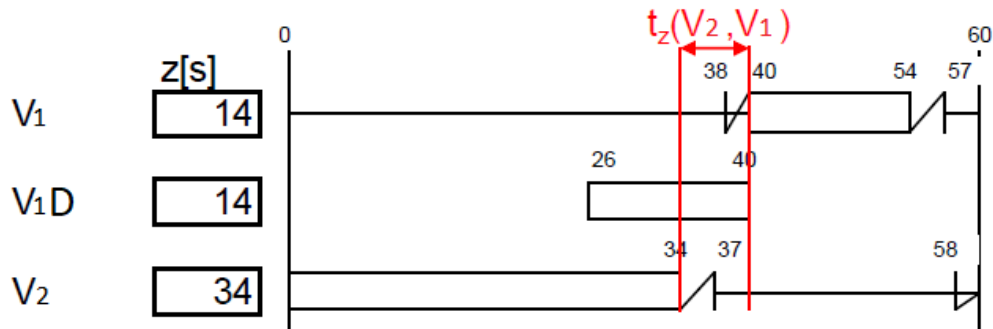
- $t_z$ - zaštitno međuvrijeme [s];
- $t_{\bar{z}}$ - trajanje žutog svjetla [s];
- $t_{c\bar{z}}$ - trajanje crveno - žutog svjetla [s].

Za pješake se koristi (5):

$$t_z \geq t_{\bar{z}} + 1 [s], \quad (5)$$

gdje je:

- $t_z$ - zaštitno međuvrijeme [s];
- $t_{\bar{z}}$ - trajanje žutog svjetla [s].



Slika 4. Prikaz zaštitnog međuvremena [5]

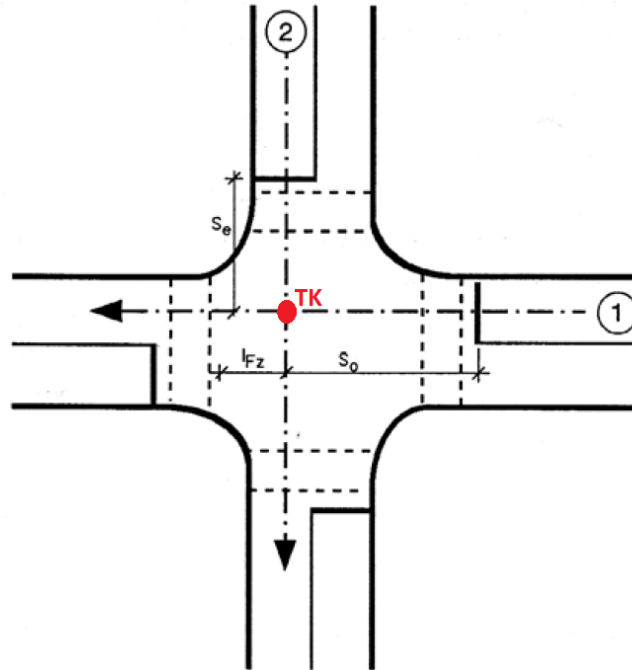
Minimalno potrebno međuvrijeme ( $t_z$ ) određuje se iz vremena prolaza ( $t_u$ ), vremena napuštanja ( $t_r$ ) i vremena ulaza ( $t_e$ ) prikazanog izrazom (6):

$$t_z = t_u + t_r + t_e, \quad (6)$$

Međuvremena se izračunavaju za sve konfliktne prometne tokove. Pritom sve grupe sudionika u prometu (pješači, biciklisti, javni gradski prijevoz (JGP), motorna vozila) treba promatrati kao odvojene tokove pa i ako se eventualno signaliziraju zajednički. Mjerodavna međuvremena za određene signalne grupe ujediniju se u matricu međuvremenu.

Za utvrđivanja međuvremena najprije treba odrediti puteve napuštanja i puteve prilaza. Kao referentne linije za utvrđivanje dužine u pravilu se koriste središnje linije voznih ili pješačkih trakova uključenih prometnih tokova.

Slika 5 prikazuje dva prometna toka (1 i 2). Put napuštanja ( $S_r$ ) sastoji se iz osnovnog puta napuštanja ( $S_0$ ) i fiktivne dužine vozila ( $l_{fz}$ ). Osnovni put napuštanja ( $S_0$ ) kod vozila je put između linije zaustavljanja i sjecišta s prilaznim putem započinjućeg prometnog toka (točka konflikta, Slika 5) [6]. Točka konflikta (TK) nalazi se na presjecištu linija putanja prometnog toka jedan i dva.



Slika 5. Prikaz slučaja konflikata [5]

#### 2.5.4. Efektivno trajanje zelenog svjetla

Vrijeme je u fazi gdje sudionici u prometu imaju slobodan prolaz. Za vozačke, tramvajske i biciklističke vozačke grupe, trajanje zelenog svjetla uvećano je za jednu sekundu dok se pješačke signalne grupe ne uvećavaju za jednu sekundu [5].

Primjer:

Vozačka signalna grupa:  $g(V_i) = z(V_i) + 1$  [s];

Pješačka signalna grupa:  $g(P_i) = z(P_i)$  [s].

Gdje je:

- $g(V_i)$  – efektivno zeleno svjetlo glavnog privoza  $V_i$  [s];
- $z(V_i)$  – osnovno zeleno svjetlo glavnog privoza [s].
- $g(P_i)$  - efektivno zeleno svjetlo pješaka  $P_i$  [s];
- $z(P_i)$  - osnovno zeleno svjetlo pješaka [s].

### 2.5.5. Prijelazna vremena

Prijelazna vremena čine žuta i crveno-žuta vremena. Žuta vremena služe za sigurno pražnjenje sudionika u raskrižju prije početka kretanja drugog prometnog toka, dok crveno-žuta vremena nagovještavaju sudionike u prometu za početak propuštanja.

#### a) Žuto vrijeme

Za reguliranje kretanje vozila promjena sa zelenog na crveno svjetlo zbog voznodinamičkih razloga prikazuje se prijelaznim signalom žuto. Prijelazno vrijeme žuto ( $t_{\frac{z}{}}$ ) određuje se prema dopuštenoj maksimalnoj brzini na prilaznom putu. Trebalo bi biti:

$$t_{\frac{z}{}} = 3 \text{ s kod } v = 50 \text{ km/h;}$$

$$t_{\frac{z}{}} = 4 \text{ s kod } v = 60 \text{ km/h;}$$

$$t_{\frac{z}{}} = 5 \text{ s kod } v = 70 \text{ km/h;}$$

$$t_{\frac{z}{}} = n \text{ s kod } v > 70 \text{ km/h.}$$

Na pojedinačnim prilazima raskrižjima vremena žutog mogu biti različitog trajanja.

#### b) Crveno i žuto vrijeme

Prijelazni signal crveno i žuto (istovremeno) prije zelenog svjetla prikazuje se kao priprema na neposredno predstojeći signal propuštanja. Prijelazno vrijeme crveno i žuto ( $t_{c\frac{z}{}}$ ) te tokove motornih vozila, ne bi smjelo biti duže od dvije sekunde (u RH 2 s).

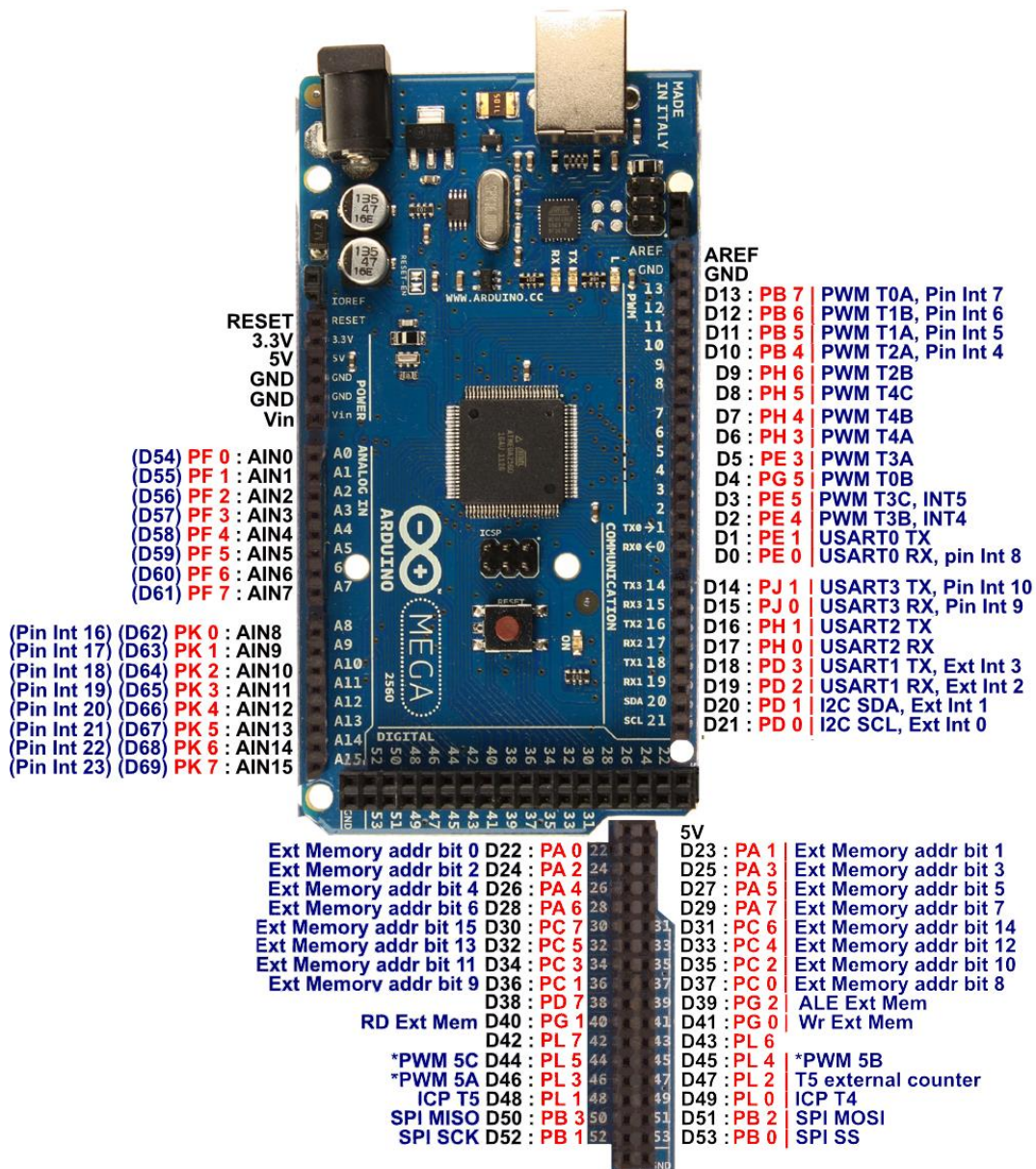


### **3. RAZVOJNO OKRUŽENJE ARDUINO**

Arduino je otvorena platforma za sve ljubitelje elektronike koji vole izrađivati različite projekte vezane za robotiku, promet i sl. Arduino može komunicirati/upravljati s tipkama, LED-icama, displejima, motorima, zvučnicima, GPS uređajima, kamerama, Internetom, pa čak i pametnim telefonom. Sastoji se od fizičke programibilne tiskane pločice (mikrokontroler) i djela programske podrške ili integriranog razvojnog okruženja (IDE, engl. Integrated Development Environment) koji se izvodi na računalu, potrebnim za pisanje i prijenos računalnog kôda fizičku ploču. Arduino platforma postala je vrlo popularna kod ljudi koji tek počinju s elektronikom. Za razliku od većine prethodno programiranih sklopovskih ploča, Arduino ne treba zasebnu sklopovsku podršku za programiranje (tzv. programator) kako bi učitao novi kôd u mikrokontroler, jednostavno se koristiti USB kabelom. Osim toga, Arduino IDE koristi pojednostavljenu verziju C++, što olakšava učenje i izradu programa.

#### **3.1 Arduino ATmega 2560 – pinovi**

Arduino Mega 2560 je tiskana pločica s mikrokontrolerom koja se zasniva na mikrokontroleru ATmega2560. Ima 54 digitalnih ulazno/izlaznih pinova od kojih se 15 može koristiti kao PWM (engl. Pulse-Width Modulation) izlaz, 16 analognih ulaza, 4 UART (sklopovska sučelja za serijsku komunikaciju), 16 MHz kristalni oscilator, USB priključak, utičnicu za napajanje, ICSP zaglavlje i tipku za resetiranje vidljivo na Slici 6. Sadrži sve što je potrebno za podršku mikrokontrolera. Jednostavno se spaja na računalo pomoću USB kabela, AC-DC adapterom ili baterijom da bi započeo rad [7].



Slika 6. Tehničke karakteristike Arduino MEGA programatora [7]

Svaku Arduino tiskanu pločicu potrebno je spojiti na izvor napajanja. On može biti USB kabel koji se priključuje na samo računalo ili vanjski istosmjerni izvor napajanja koji je preporučljiv da bude između 6 i 12V. Arduino MEGA koristi niz pinova koji se dijele na naponske (5V; 3,3V i GND), analogne i digitalne ulazno/izlazne pinove i PWM te ARF pinove.

**GND:** Masa – spajanje „kratko na zemlju“. Na Arduinu MEGA postoji 5 GND pinova, od kojih se bilo koja od njih može koristiti za uzemljenje kruga.

**5V i 3,3V:** Napajanje – naponski pinovi koji osiguravaju 5V i 3,3V napon za krugove.

**ANALOG IN:** Analogni pinovi koji čitaju signale iz analognog senzora (poput senzora temperature) i pretvaraju ga u digitalnu vrijednost koja se može pročitati.

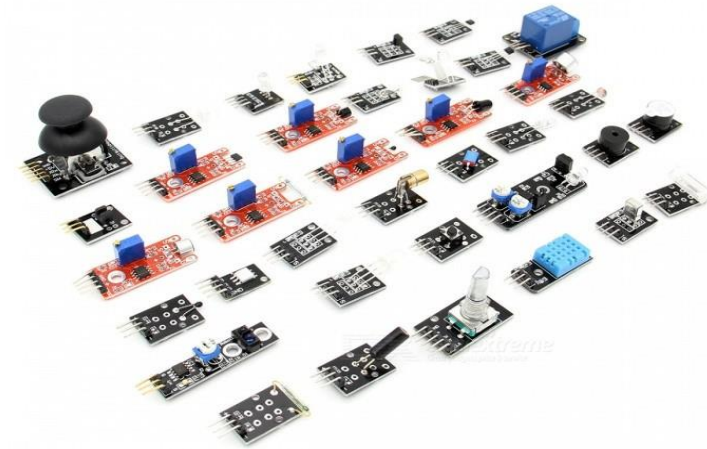
**DIGITAL:** Ovi pinovi mogu koristiti kao digitalni ulaz (npr. pritisak na gumb) i digitalni izlaz (npr. napajanje LED diode).

**PWM:** Pinovi za dobivanje analognih rezultata digitalnim ulazima. Signal se modulira kako bi na izlazu PWM pinova bio moguć signal (napon) u rasponu od 0v do 5v. Koristi se za razne primjene, npr. razina jačine svijetljenja LED-ice, upravljanje brzine vrtnje motora, itd.

**AREF:** Pinovi koji se koriste za postavljanje vanjskog referentnog napona (između 0 i 5 V) kao gornje granice za analogne ulazne pinove.

### 3.2. Arduino osjetila

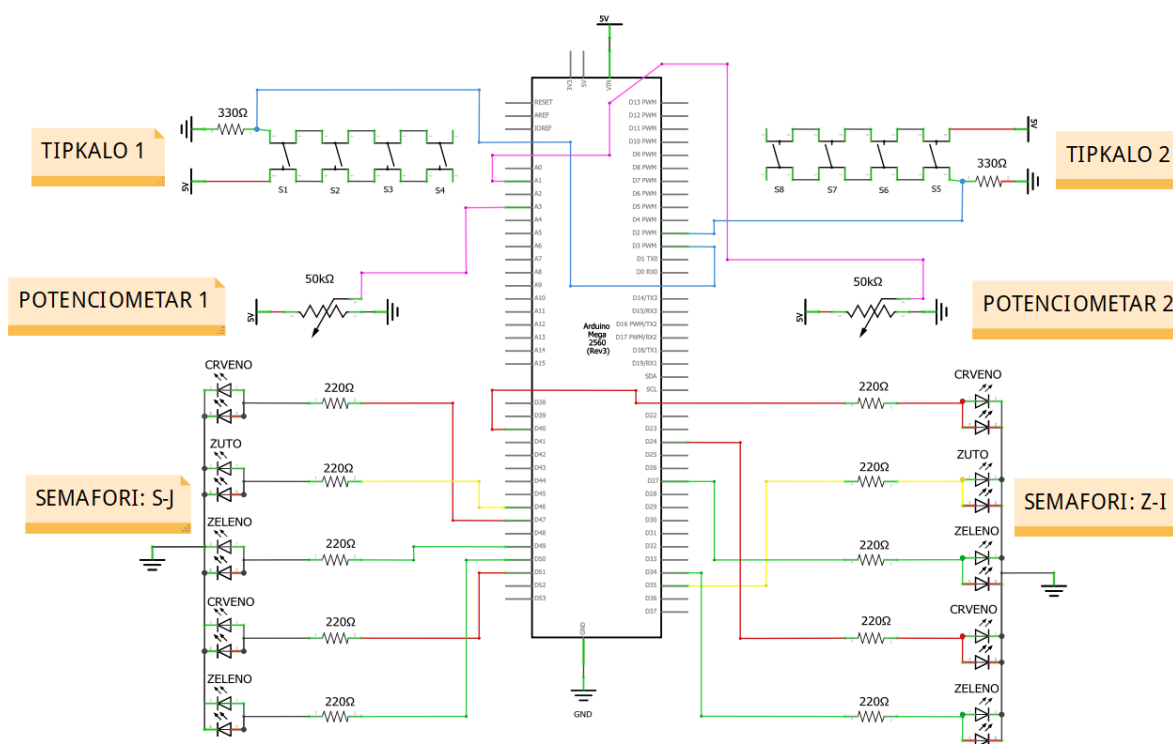
Arduino s jednostavnim programskim kôdom može kontrolirati i komunicirati s velikim brojem osjetila (Slika7) kao što su: osjetilo svjetlosti, temperature, tlaka, udaljenosti, ubrzanja, ugljičnog monoksida, radioaktivnosti, vlažnosti, barometarskog tlaka, visine tekućine i tako dalje.



Slika 7. Osjetila kompatibilna s Arduino platformom [8]

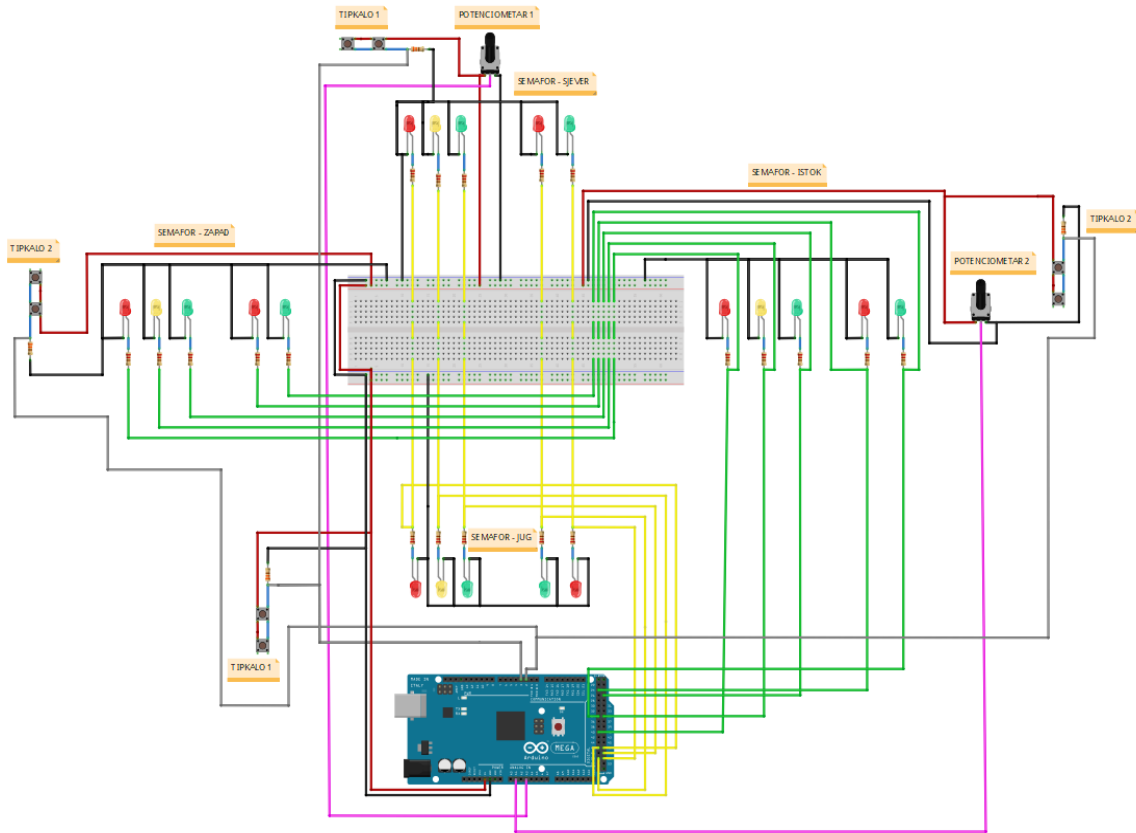
## 4. SKLOP ZA SIMULACIJU SEMAFORIZIRANOG RASKRIŽJA

Sklop za upravljanje semaforom sastavljen je od makete raskrižja, Arduino razvojne platforme, kućišta, semafora, potencijometra i tipkala. Slika 8 prikazuje shemu spajanja 4 svjetlosne signalizacije (semafori), 8 tipkala za pješake i dva potencijometra za simulaciju protoka vozila pomoću Arduina. Kada je sklop izrađen i programski kôd učitana u Arduino programsku pločicu, kreće izvršavanje programskog kôda, odnosno kreće izmjena signalnih grupa i provjera uvjeta pješačkih tipkala i potencijometra za simulaciju protoka vozila.



Slika 8. Elektronička shema sklopa za simulaciju semaforiziranog raskrižja

Maketa komponenti (Slika 9) služi za vizualnu percepciju sklopa sustava. Sačinjena je od četiri svjetlosne signalizacije (semafora) raspoređene po stranama svijeta (raskrižje s 4 prometna traka), potencijometrima (dva) koji simuliraju gustoću vozila na privozu S-J i Z-I te od 8 tipkala raspoređena tako da svaki semafor posjeduje dva tipkala.

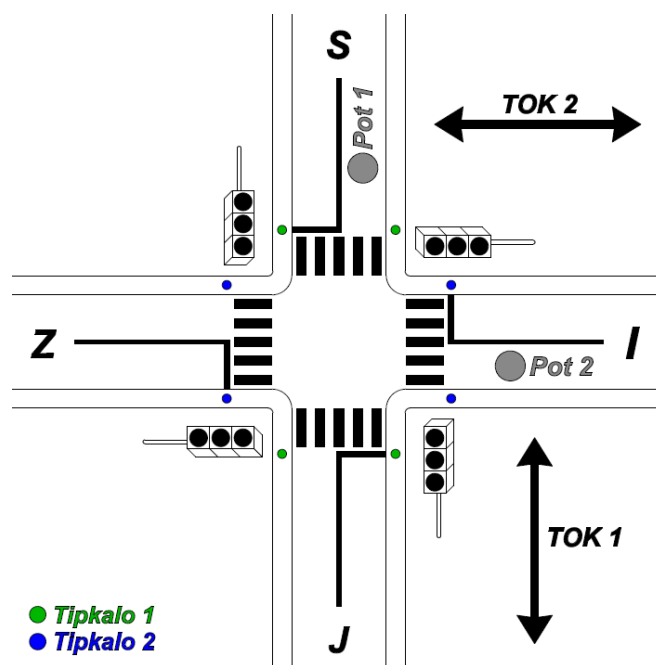


Slika 9. Eksperimentalna maketa komponenti semaforiziranog raskrižja

#### 4.1. Princip rada semaforiziranog raskrižja

Semaforiziranim raskrižjem upravlja isključivo program, odnosno kôd koji izvršava signalni plan semafora. U prvom obliku, programski kôd je pisan za fiksno upravljanje semaforizacijom zbog jednostavnosti i razumijevanja, gdje mu je naknadno dodana mogućnost adaptivnog upravljanja. Naime, kompletan kôd zasnovan je na provjeravanju uvjeta dolaska pješaka (pritisnuto tipkalo) i usporedbom prometnog toka na privozima S-J i Z-I (omjera potenciometra 1 i 2). Pomoću ta dva ulazna parametra, sklop za upravljanje semaforiziranim raskrižjem obavlja naredbe koje dolaze iz napisanog programskog kôda.

U daljnjem opisu, Slika 10 prikazuje razmještaj tipkala i potenciometra s obzirom na strane svijeta gdje će biti lakše razumjeti ukupnu ideju o simulaciji adaptivnim upravljanjem semaforiziranim raskrižjem. Potenciometri imaju ulogu simuliranja gustoće prometnog toka. U radu simulacije semaforiziranog raskrižja korištena su dva potenciometra, a ne četiri kao što bi bilo logičnije iz razloga jednostavnosti izvedbe i razumijevanja cijele problematike. Potenciometar jedan simulira protok vozila u smjeru S-J i J-S, dok potenciometar dva simulira protok vozila u smjeru Z-I i I-Z. Naime, opseg svakog potenciometra podijeljen je na tri djela. Od 0% - 33,3%, 33,3% - 66,6% i od 66,6% - 100%. U prvom djelu opsega, protok vozila je najmanji od maksimalnog, srednji opseg govori da je protok vozila srednje vrijednosti od maksimalne, dok zadnji dio opsega govori da je protok vozila na tom smjeru odvijanja prometa maksimalan. Dodavši na sve to još jedan zasebni potenciometar, dobije se devet mogućih kombinacija za produljenje, odnosno skraćivanje trajanja zelenog svjetla. Trajanje ciklusa je konstantno dok se faze jedan i dva produljuju, odnosno skraćuju. Omjer razlike potenciometra jedan i dva ukazuje na vrijeme produljenja/smanjenja određenog privoza prikazano u tablici 2.



Slika 10. Prikaz makete raskrižja u tlocrtu s označenim prometnim tokovima, semaforima te tipkalima za pješake

Zbog lakšeg razumijevanja, iskoristit će se jedan jednostavni primjer. Pretpostavimo da je u smjeru S-J protok vozila srednje vrijednosti, odnosno gledano na potencionometru, to je opseg dva. Za smjer Z-I uzmimo pretpostavku da je prometna potražnja maksimalna, odnosno gledano na potencionometru, to je opseg 3. Razlika od opsega 2 i 3 je 1, što znači da će produljenje zelenog svjetla biti za 1 sekundu za smjer Z-I, tj. Pogledavši u tablicu 2, za vrijednosti opsega 1 i 2, vidimo da je razlika u trajanju zelenog svjetla jednaka jednoj sekundi.

Tablica 2. Produljenje trajanja zelenog svjetla

<b>OMJER POTENCIOMETRA 1 I 2 PREMA OPSEGU [POT1 – POT2]</b>	<b>RAZLIKA U VREMENU TRAJANJA ZELENOG SVJETLA [S]</b>
<b>1-1</b>	0s
<b>1-2</b>	1s
<b>1-3</b>	2s
<b>2-1</b>	1s
<b>2-2</b>	0s
<b>2-3</b>	1s
<b>3-1</b>	2s
<b>3-2</b>	1s
<b>3-3</b>	0s

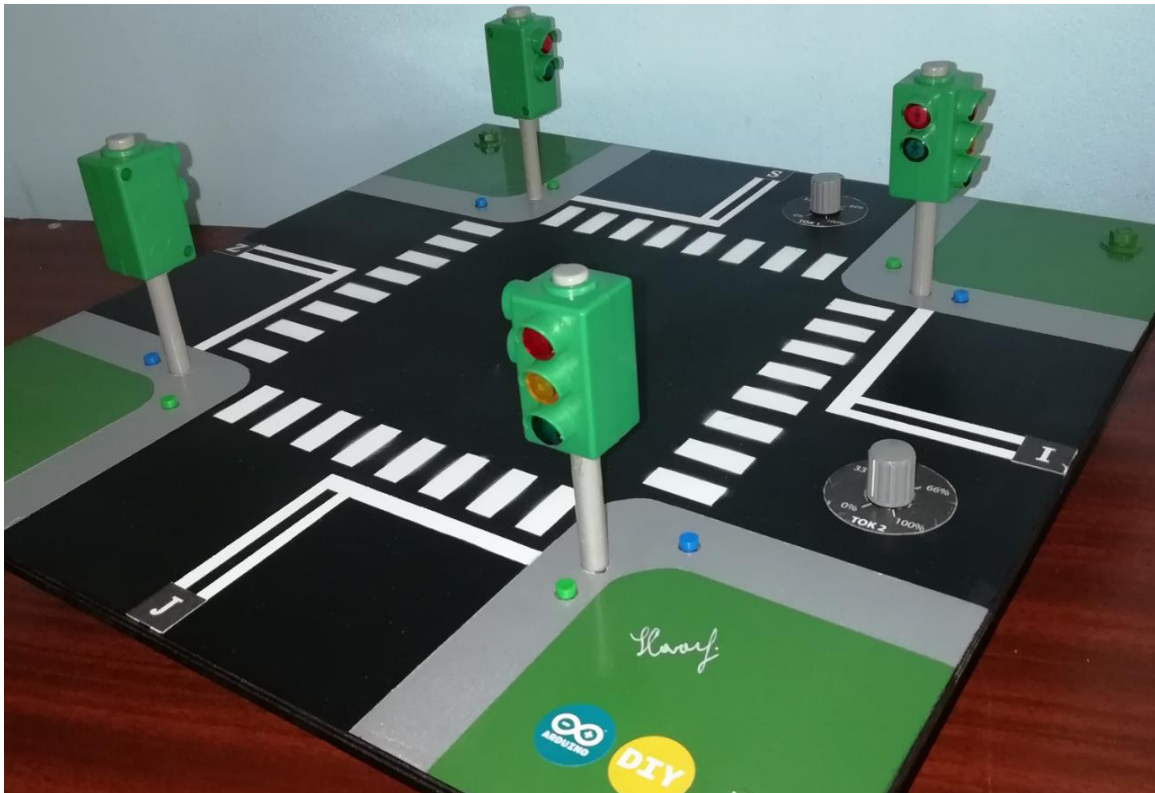
Tipkala imaju ulogu simuliranja dolaska pješaka. U radu simulacije semaforiziranog raskrižja korišteno je 8 tipkala. Dva tipkala po svakom privozu daje ukupni broj tipkala. Svih osam tipkala može se promatrati kao dva nezavisna tipkala, što znači da imamo dva puta četiri



tipkala spojena u paralelni spoj iz razloga ako pješak pritisne jedno od dva tipkala. Npr. Na privozu S-J, nakon nekog vremena na tom glavnom privozu morat će se zeleno svjetlo prebaciti u crveno. Crveno svjetlo bit će na privozu S-J i J-S, što povlači sljedeću činjenicu. Ako se pritisne bilo koje tipkalo od ta četiri koja su na jednom od dva smjera prometovanja, ishod je uvijek isti. Od tuda razmatranje „osam tipkala se svodi na samo dva“.

#### **4.2. Izrada makete semaforiziranog raskrižja**

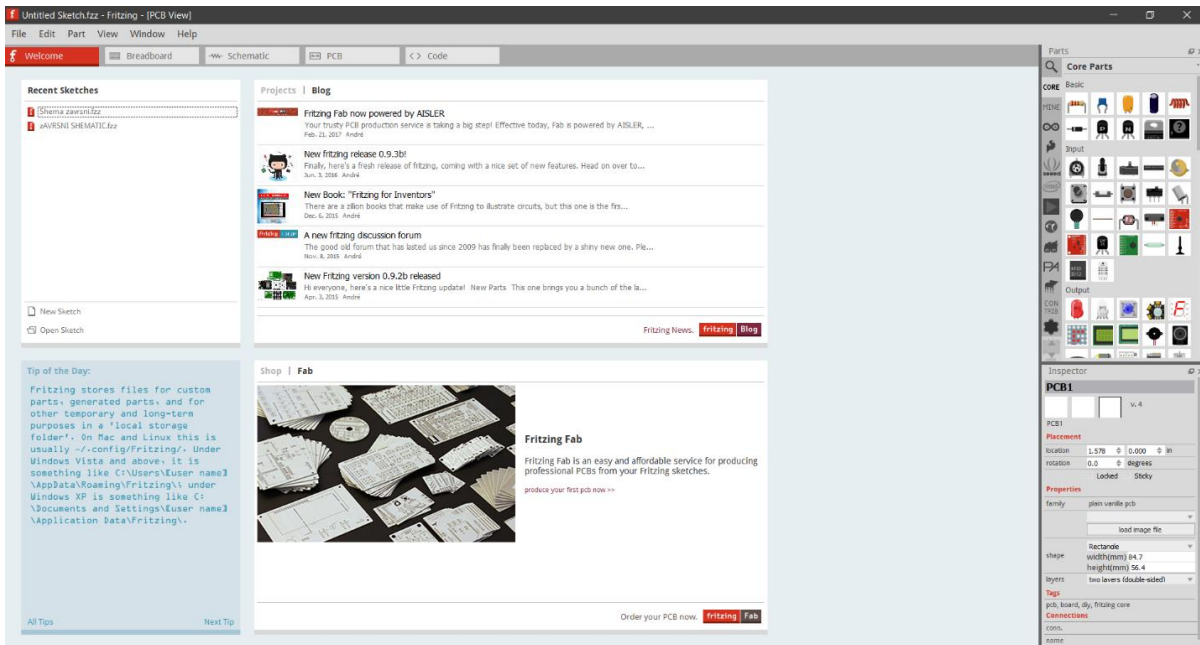
Maketa semaforiziranog raskrižja sastoji se od drvenog postolja (šperploča), semafora, tipkala, potencijometra, Arduino programatora, nogica, vodiča i kućišta programatora. Cijeli proces izrade možemo svrstati u više faza izrade. Izrada je krenula od projektiranja na papiru, preciznog skiciranja te kotiranja na samome postolju. Iduća faza bila je faza bušenja rupa za semafore, tipkala, potencijometre i nožice. Nakon toga, kreće lijepljenje, obrezivanje trake i bojanje. Samo bojanje, odvijalo se u 4 navrata zbog različitih segmenata na postolju. Iduća faza bila je montiranje potencijometra, tipkala, semafora, nožica, kućišta programatora, sami programator Arduino u kućište te lijepljenje naljepnica za strane svijeta i vrijednosti potencijometra. Na poslijetku je trebalo još povezati sve komponente s Arduino programatorom i učitati programski kôd u njegovu memoriju, gdje je sve bilo spremno za testiranje (Slika 11).



Slika 11. Maketa adaptivno upravljivog semaforiziranog raskrižja

#### 4.2.1. Izrada dokumentacije

Elektronička shema kao i maketa komponenti izrađeni su u programskoj podršci pod nazivom Fritzing (Slika 12). To je programska podrška otvorenog kôda za ljude koji nisu inženjeri. Savršen je alat za dizajnere, inovatore, hobiste i nastavnike za kreiranje prototipa ili čak izrade PCB pločica. Jako je jednostavan za korištenje i lako se uči raditi u njemu. Postoje mape s raznim mikrokontrolerima, elektroničkim komponentama i alatima, što ga čini jako zanimljivim.



Slika 12. Izgled sučelja programse podrške Fritzing

Fritzing se sastoji od kartica Welcome, Breadboard, Schematic, PCB i Code koji predstavljaju glavnu navigaciju Fritzing-om. Welcome je početna kartica na kojem se nalaze posljednje novosti. Kartica Breadboard omogućuje pogled na eksperimentalnu pločicu, odnosno daje osjećaj fizičkog slaganja komponenti. Kartica Schematic je alat za kreiranje elektroničkih shema. Komponente i module koje koristimo u jednom od ovih kartica, automatski se nalaze i u drugim. Kartica PCB služi za kreiranje tiskanih elektroničkih pločica. Na posljetku kartica Code služi za pisanje i mijenjanje programskog kôda te njegovo učitavanje izravno na Arduino pločicu [9].

#### 4.2.2. Moguće nadogradnje

Ovu maketu za simulaciju semaforiziranog raskrižja moguće je nadograditi. Za bolju preciznost rada sustava, moguće je dodati još dva potenciometra te doraditi programski kôd na osnovi 4 potenciometra. Iduća nadogradnja bila bi postavljanje senzora (osjetila) udaljenosti (Slika 13) umjesto potenciometra, gdje bi s programskim kôd-om upravljali senzorom koji bi brojao promet na svakom privozu. Naposljetku bi programski kôd sa svim ulaznim

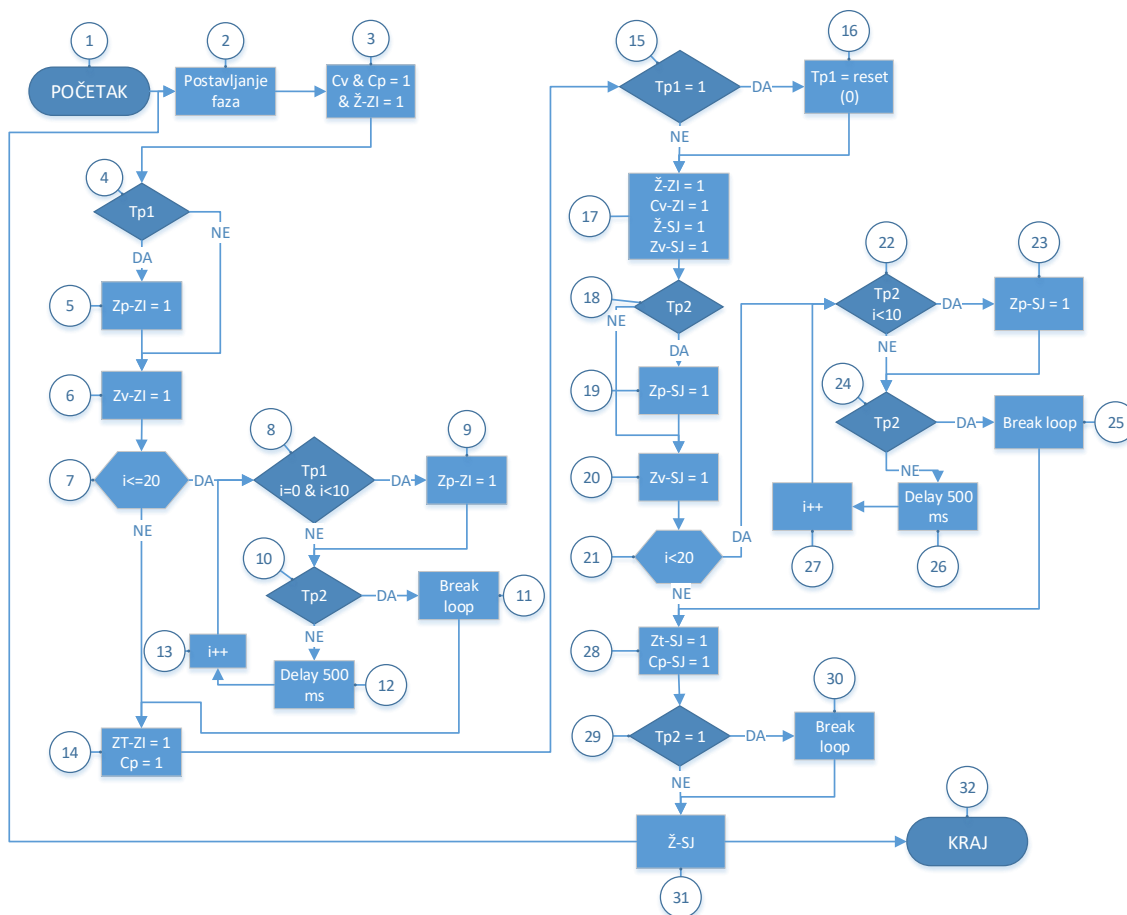
parametrima sa senzora udaljenosti odlučivao o produljenju/skraćenju trajanja zelenog svjetla za pojedinu fazu.



Slika 13. Osjetilo za mjerenje udaljenosti zasnovano na ultrazvuku [10]

## 5. PROGRAM ZA UPRAVLJANJE SEMAFOROM

U ovom poglavlju biti će prikazana logika rada napravljene programske podrške kroz prikaz dijagrama toka (Slika14) i programskog kôda priloženog u prilogu 1.



Slika 14. Dijagram toka rada adaptivno upravljivog semaforiziranog raskrižja

Opis izvršavanja dijagrama toka prema blokovima označenim u dijagramu toka prikazanog na slici 14 je dan u nastavku.

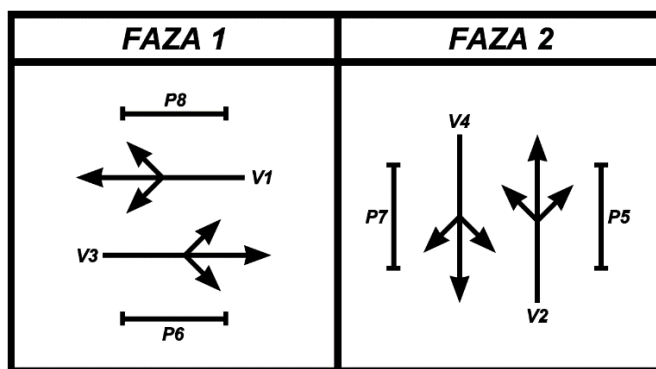
1. Uključivanje napajanja.
2. Postavljanje faza.
3. Crveno svjetlo za sve pješake i vozila postavlja se u stanje „1“ u trajanju od 3 s, zatim se uključuje žuto svjetlo u trajanju od 2 s.

4. Ispituje se aktivacija tipkala 1. Ako nije aktivirano, ne uključuje zeleno svjetlo pješacima na tom privozu.
5. Ako je tipkalo 1 aktivirano, uključuje se zeleno svjetlo pješacima na tom privozu.
6. Uključuje se zeleno svjetlo vozačkoj signalnoj grupi istog smjera (Z-I).
7. Petlja koja se vrti 20 puta po 500 ms, tj. 10 s i koja produljuje ili skraćuje trajanje zelenog svjetla na privozu Z-I.
8. Ispituje se aktivacija tipkala 1 svakih 500 ms za vrijeme trajanja prve polovice zelenog svjetla za vozila.
9. U slučaju pritisnutog tipkala, pušta se pješake na privozu Z-I.
10. Ispituje se aktivacija tipkala 2.
11. Ako je pritisnuto tipkalo 2, prelazi se na fazu 2, odnosno gasi se zeleno svjetlo vozila privoza Z-I i pušta se pješake S-J nakon 2 s od kraja zelenog svjetla vozača.
12. Ako tipkalo 2 nije aktivirano, provjerava se aktivacija opet nakon 500 ms i tako 10 puta.
13. Varijabla „i“ je brojač petlje.
14. Gasi se trepćuće zeleno svjetlo u smjeru Z-I i uključuje crveno svjetlo za pješake.
15. Ispituje se aktivacija tipkala 1.
16. Ako je tipkalo 1 aktivirano, resetira se.
17. Uključuje se žuto u trajanju od 2 s pa crveno svjetlo vozačke signalne grupe u trajanju od 2 s, te se uključuje žuto svjetlo u trajanju od 2 s, zatim zeleno vozačke signalne grupe privoza S-J.
18. Ispitivanje aktivacije tipkala 2.
19. Ako je, uključuje se zeleno svjetlo pješacima na tom privozu
20. Uključuje se zeleno svjetlo za vozila na privozu S-J.

21. Petlja koja se vrti 20 puta po 500 ms, tj. 10 s i koja produljuje ili skraćuje trajanje zelenog svjetla na privozu S-J.
22. Ispituje se aktivacija tipkala 2 svakih 500 ms za vrijeme trajanja prve polovice zelenog svjetla za vozila.
23. Ako je tipkalo aktivirano, uključuje se zeleno svjetlo pješacima na privozu S-J.
24. Ispituje se aktivacija tipkala 2 u drugoj polovici trajanja zelenog svjetla.
25. Ako je aktivirano, petlja se prekida.
26. Ako tipkalo 2 nije aktivirano, provjerava se aktivacija svakih 500 ms i tako 10 puta.
27. Varijabla „i“ je brojač petlje.
28. Gašenje trepćućeg zelenog svjetla u smjeru S-J i uključivanje crvenog svjetla za pješake.
29. Ispituje se aktivacija tipkala 2 u drugoj polovici trajanja zelenog svjetla.
30. Ako je uvjet zadovoljen, petlja se prekida.
31. Ako uvjet nije zadovoljen, aktivira se žuto svjetlo na privozu S-J.
32. Ciklus kreće iz ponova.

## 6. TESTIRANJE SIMULATORA

Signalni plan (Slika 16), tj. plan izmjene signala je pregled trajanja svjetlosnih signalnih pojmova. Sastoji se od prikaza duljine trajanja efektivnog zelenog svjetla, promjenjivog trajanja zelenog svjetla i trepćućeg zelenog svjetla, duljine ciklusa, faza, broja faza, signalnog pojma te zaštitnog međuvremena. Signalni plan simulacijskog modela sastoji se od dvije faze prikazanih na Slici 15.



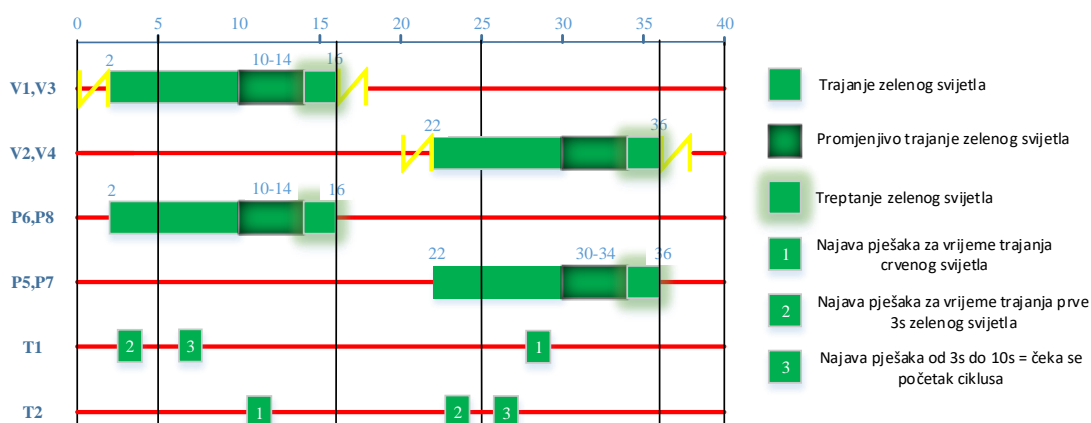
Slika 15. Pregled faza simulacijskog modela

U semaforiziranom raskrižju dok nema najave pješaka, na pješačkim prijelazima je pripadni signalni pojam konstantno u crvenom. U slučaju kada pješak pošalje zahtjev za prijelazom prometnice, zeleno svjetlo za pješake uključuje se prema tri slučaja kao što je prikazano na signalnom planu (Slika 16). U prvom slučaju, najava pješaka dešava se u vremenu trajanja crvenog svjetla na glavnom privozu. Dok je tako, zeleno svjetlo za pješake uključuje se istovremeno sa zelenim svjetlom na glavnom privozu. U drugom slučaju zeleno svjetlo za pješake bit će uključeno u trenutku pritiska na tipkalo ako pješak najavi svoje prisustvo u prve tri sekunde trajanja zelenog svjetla na glavnom privozu jer minimalno zeleno svjetlo traje 8 sekundi plus dvije sekunde trepćućeg zelenog svjetla što je ukupno 10 sekundi, a izračunom, mjerenjem i iz prijašnjih spoznaja znamo da je vrijeme napuštanja raskrižja pješaka maksimalno 7 sekundi. Ta brojka od 7 sekundi uključuje svu životnu dob ljudi (ljudi starije dobi, djeca, invalidi i itd.) kako bi svi pješaci sigurno napustili pješački prijelaz. Na kraju, treći slučaj najave pješaka odvija se u vremenskom intervalu od treće sekunde pa do kraja trajanja zelenog svjetla



glavnog privoza. Tada se pješačko zeleno svjetlo aktivira na početku idućeg ciklusa u fazi gdje je aktivirano tipkalo za najavu pješaka.

Adaptivno upravljanje semaforiziranim raskrižjem upravlja se položajem potenciometra jedan i dva. Naime, ciklus je konstantan i iznosi 40 sekundi. Trajanje zelenog svjetla (bez trepćućeg zelenog svjetla) iznosi od 8 do 12 sekundi plus 2 sekunde trepćućeg zelenog svjetla ovisno o prometnoj potražnji. Dok su potenciometri jedan i dva u položaju istog opsega, trajanje zelenog svjetla iznosit će 10 sekundi plus 2 sekunde trepćućeg zelenog svjetla. To je referentna vrijednost trajanja zelenog svjetla od koje se duljina trajanja može povećati i smanjiti za dvije sekunde.



Slika 16. Signalni plan

Maketa simulacije semaforiziranog raskrižja prikazana je na Slici 17 kao model raskrižja u realnom svijetu. Na modelu je prikazana sva potrebna tehnička podrška za rad raskrižja. Tu su semafori, kamere za detekciju vozila na pojedinim privozima te tipkala za najavu pješaka.



Slika 17. Model jednostavnog adaptivno upravljivog semaforiziranog raskrižja u realnom svijetu

## 7. ZAKLJUČAK

Danas, iz dana u dan raste prometna potražnja u svim oblicima prometa pa tako i u cestovnom. Naročito u gusto naseljenim urbanim sredinama. Da bi se zadovoljio volumen potražnje, izgrađuju se nove i proširuju stare prometnice. Zato je potrebno posebno voditi računa o sigurnosti odvijanja prometa i efikasnosti sustava te se iz tog razloga javlja potreba za korištenjem novih tehnologija kojima bi se ta količina prometne potražnje mogla nadzirati i upravljati.

Kod upravljanja semaforiziranim raskrižjem, sustav adaptivnog upravljanja pokazao se nužnim u svim većim gradskim semaforiziranim raskrižjima. Sustav adaptivnog upravljanja promatramo kao jednu granu ITS-a punu naprednih tehnoloških rješenja za postizanje veće sigurnosti i učinkovitosti u odvijanju prometa kroz semaforizirana raskrižja. Na mjestima gdje je ovaj sustav implementiran, prometna zagušenja i vremena čekanja su znatno smanjena.

U ovom radu je primijenjena Arduino razvojna platforma za razvoj i implementaciju simulacije sustava adaptivnog upravljanja semaforiziranim raskrižjem. Ta razvojna platforma idealna je za korištenje u sustavima upravljanja cestovnog prometa zbog svojih tehničkih karakteristika. Kao što je simulacija modela pokazala, adaptivno upravljivo semaforizirano raskrižje pridonosi prilagođavanju prometne potražnje. Faze se produljuju i skraćuju s obzirom na gustoću prometa na privozima, a najava pješaka uz određene uvjete upućuje na aktivaciju zelenog pješačkog svijetla, gdje u suprotnom svijetli crveno svijetlo zbog ne-najave pješaka i bržeg prolaska skretača.

Moguća nadogradnja postojećeg sustava bila bi grafičko sučelje za nadzor i upravljanje jednostavnim semaforiziranim raskrižjem jer Arduino platforma ima mogućnost serijske komunikacije sa računalom. Druga nadogradnja bila bi znatno složeniji algoritam koji bi upravljao istim. Kooperativno upravljanje prometom predstavlja nadgradnju adaptivnom upravljanju sustava i kao takav bio bi treća nadogradnja semaforiziranog raskrižja. Kooperativno upravljanje sustavom u cestovnom prometu predstavlja međusobnu komunikaciju vozača, vozila i infrastrukture čime se upravlja cjelovitim sustavom, a ne samo

infrastrukturu kao što je to slučaj adaptivnog upravljanja prometom. To bi se postiglo implementacijom Bluetooth tehnologije na prometnici i vozilu gdje bi oni međusobno komunicirali.

## LITERATURA

- [1] Legac, I.: *Raskrižja javnih cesta*, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, 2008, str. 69.
- [2] Bošnjak, I.: *Inteligentni transportni sustavi*, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, 2006, str. 200.
- [3] Vujić, M.: Strategije i taktike upravljanja prometom, PowerPoint prezentacija, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, slajd 3
- [4] Vugrinec, F.: Mogućnost primjene Arduino razvojne platforme za simulaciju jednostavnog semaforiziranog pješačkog prijelaza, Završni rad, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, 2017, str. 6.
- [5] Lanović, Z.: Signalni plan, Cestovna telematika, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, 2016.
- [6] Novačko, L.; Pilko, H.: Cestovne prometnice II, Fakultet prometnih znanosti, Sveučilište u Zagrebu, 2017, str. 26.
- [7] Arduino Store, URL: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>, 15.08.2018.
- [8] Deal extreme gadgets, URL: <http://www.dx.com/p/arduno-37-in-1-sensor-module-kit-black-142834#.W3Qh6egzY2w>, 15.08.2018.
- [9] E-radionica.com, Fritzing, URL: <https://e-radionica.com/hr/blog/2016/02/14/fritzing/>, 15.08.2018.
- [10] Mini in the Box, URL: [https://www.miniinthebox.com/hr/p/hc-sr04-ultrazvucni-senzor-za-mjerenje-udaljenosti-modul-za-arduino\\_p1764217.html](https://www.miniinthebox.com/hr/p/hc-sr04-ultrazvucni-senzor-za-mjerenje-udaljenosti-modul-za-arduino_p1764217.html), 15.08.2018.

## POPIS SLIKA

Slika 1. Prikaz detekcije vozila [3] .....	7
Slika 2. Pregled signalnog plana [5] .....	12
Slika 3. Pregled faza [5] .....	13
Slika 4. Prikaz zaštitnog međuvremena [5] .....	15
Slika 5. Prikaz slučaja konflikata [5] .....	16
Slika 6. Tehničke karakteristike Arduino MEGA programatora [7] .....	19
Slika 7. Osjetila kompatibilna s Arduino platformom [8] .....	21
Slika 8. Elektronička shema sklopa za simulaciju semaforiziranog raskrižja .....	22
Slika 9. Eksperimentalna maketa komponenti semaforiziranog raskrižja .....	23
Slika 10. Prikaz makete raskrižja u tlocrtu s označenim prometnim tokovima, semaforima te tipkalima za pješake .....	24
Slika 11. Maketa adaptivno upravljivog semaforiziranog raskrižja .....	27
Slika 12. Izgled sučelja programse podrške Fritzing .....	28
Slika 13. Osjetilo za mjerenje udaljenosti zasnovano na ultrazvuku [10] .....	29
Slika 14. Dijagram toka rada adaptivno upravljivog semaforiziranog raskrižja .....	30
Slika 15. Pregled faza simulacijskog modela .....	33
Slika 16. Signalni plan .....	34
Slika 17. Model adaptivno upravljivog semaforiziranog raskrižja u realnom svijetu .....	35

## POPIS TABLICA

Tablica 1. Ulazne veličine za analizu semaforiziranog raskrižja [1].....	6
Tablica 2. Produljenje trajanja zelenog svjetla.....	25

## POPIS FORMULA

- (1) Ciklus
- (2) Ciklus
- (3) Faza
- (4) Zaštitno međuvrijeme za vozila
- (5) Zaštitno međuvrijeme za pješake
- (6) Zaštitno međuvrijeme



## POPIS KRATICA

Cv	crveno svjetlo za vozila
Cp	crveno svjetlo za pješake
Ž-ZI	žuto svjetlo na privozu Z-I
Ž-SJ	žuto svjetlo na privozu S-J
Tp1	tipkalo 1 za pješake
Tp2	tipkalo 2 za pješake
Zt-SJ	treptajuće zeleno svjetlo na privozu S-J
Zt-ZI	treptajuće zeleno svjetlo na privozu Z-I
Zp-ZI	zeleno svjetlo za pješake na privozu Z-I
Zv-ZI	zeleno svjetlo za vozila na privozu Z-I
Zp-SJ	zeleno svjetlo za pješake na privozu S-J
Zv-SJ	zeleno svjetlo za vozila na privozu S-J
Cv-ZI	crveno svjetlo za vozila na privozu Z-I
Cp-SJ	crveno svjetlo za vozila na privozu S-J

## PRILOG 1. PROGRAMSKI KÔD ZA UPRAVLJANJE SEMAFORIZIRANIM RASRIŽJEM

// Inicijalizacija varijabli.

```
int r1 = 47;  
int y1 = 46;  
int g1 = 49;
```

```
int r2 = 51;  
int g2 = 50;
```

```
int r3 = 40;  
int y3 = 35;  
int g3 = 27;
```

```
int r4 = 24;  
int g4 = 34 ;
```

```
int fazaZI = 1;  
int fazaSJ = 1;  
int faza1 = 1;  
int faza2 = 1;
```

```
bool tipkalo1;  
bool tipkalo2;  
bool pj1=false;  
bool pj2=false;  
int i=0;
```

// Metode kojima se pale i gase svjetala na semaforima.

```
void red1(bool state) {  
    digitalWrite(r1, state);  
    digitalWrite(g1, !state);  
    digitalWrite(y1, !state);  
}  
void green1(bool state) {  
    digitalWrite(r1, !state);  
    digitalWrite(g1, state);  
    digitalWrite(y1, !state);  
}  
void yellow1(bool state) {  
    digitalWrite(r1, !state);  
    digitalWrite(g1, !state);
```

```

    digitalWrite(y1, state);
}
void yellow1X(bool state) {
    digitalWrite(y1, state);
}
void red2(bool state) {
    digitalWrite(r2, state);
    digitalWrite(g2, !state);
}
void green2(bool state) {
    digitalWrite(r2, !state);
    digitalWrite(g2, state);
}
void yellow3(bool state) {
    digitalWrite(r3, !state);
    digitalWrite(g3, !state);
    digitalWrite(y3, state);
}
void red3(bool state) {
    digitalWrite(r3, state);
    digitalWrite(g3, !state);
    digitalWrite(y3, !state);
}
void green3(bool state) {
    digitalWrite(r3, !state);
    digitalWrite(g3, state);
    digitalWrite(y3, !state);
}
void yellowX3(bool state) {
    digitalWrite(y3, state);
}
void green4(bool state) {
    digitalWrite(r4, !state);
    digitalWrite(g4, state);
}
void red4(bool state) {
    digitalWrite(r4, state);
    digitalWrite(g4, !state);
}
// Metode treptanja zelenih svjetala.
void blinkanjeZelenogSJ(bool state) {
    digitalWrite(g1, !state);
    if(pj2)
        digitalWrite(g4, !state);
}

```

```

delay(500);
digitalWrite(g1 , state);
if(pj2)
  digitalWrite(g4 , state);
delay(500);
digitalWrite(g1 , !state);
if(pj2)
  digitalWrite(g4 , !state);
delay(500);
digitalWrite(g1 , state);
if(pj2)
  digitalWrite(g4 , state);
delay(500);
digitalWrite(g1 , !state);
if(pj2)
  digitalWrite(g4 , !state);
delay(500);
digitalWrite(g1 , state);
if(pj2)
  digitalWrite(g4 , state);
}
void blinkanjeZelenogZI(bool state) {
digitalWrite(g3 , !state);
if(pj1)
  digitalWrite(g2 , !state);
delay(500);
digitalWrite(g3 , state);
if(pj1)
  digitalWrite(g2 , state);
delay(500);
digitalWrite(g3 , !state);
if(pj1)
  digitalWrite(g2 , !state);
delay(500);
digitalWrite(g3 , state);
if(pj1)
  digitalWrite(g2 , state);
delay(500);
digitalWrite(g3 , !state);
if(pj1)
  digitalWrite(g2 , !state);
delay(500);
digitalWrite(g3 , state);
if(pj1)

```

```

    digitalWrite(g2 , state);
}
// Interrupt metode koje postavljaju zastavicu koja označava da je tipkalo pritisnuto.
void tip1(){
    tipkalo1 = true;
    Serial.println("Stisnuto tipkalo 1");
}
void tip2(){
    tipkalo2 = true;
    Serial.println("Stisnuto tipkalo 2");
}
// Defaultna setup funkcija Arduina koja se izvršava pri pokretanju.
// Postavljanje pinova ulaz/izlaz.
void setup() {
    pinMode(r1, OUTPUT);
    pinMode(y1, OUTPUT);
    pinMode(g1, OUTPUT);

    pinMode(r2, OUTPUT);
    pinMode(g2, OUTPUT);
    pinMode(r3, OUTPUT);
    pinMode(y3, OUTPUT);
    pinMode(g3, OUTPUT);

    pinMode(r4, OUTPUT);
    pinMode(g4, OUTPUT);

    pinMode(2, INPUT_PULLUP);
    pinMode(3, INPUT_PULLUP);
    // Postavljanje interrupt metode na pinove 2 i 3.
    // Metode postavljaju zastavice.
    attachInterrupt(digitalPinToInterrupt(2), tip1, RISING);
    attachInterrupt(digitalPinToInterrupt(3), tip2, RISING);
    Serial.begin(9600);
}
// Petlja sa beskonačnim brojem ponavljanja.
void loop() {
    // Čitanje stanja potenciometra.
    int pot1 = analogRead(3);
    faza1 = map(pot1, 0, 1023, 1, 3);

    int pot2 = analogRead(1);
    faza2 = map(pot2, 0, 1023, 1, 3);
    // Provjeravanje uvjeta opsega potenciometra.

```

```
// Dodjeljivanje ili oduzimanje zelenog svjetla s obzirom na položaj potencijetra.
```

```
if (faza1==1 && faza2==1){  
    fazaSJ=0;  
    fazaZI=0;  
}  
else if(faza1==1 && faza2==2){  
    fazaSJ=-2;  
    fazaZI=2;  
}  
else if(faza1==1 && faza2==3){  
    fazaSJ=-4;  
    fazaZI=4;  
}  
else if(faza1==2 && faza2==1){  
    fazaSJ=2;  
    fazaZI=-2;  
}  
else if(faza1==2 && faza2==2){  
    fazaSJ=0;  
    fazaZI=0;  
}  
else if(faza1==2 && faza2==3){  
    fazaSJ=-2;  
    fazaZI=2;  
}  
else if(faza1==3 && faza2==1){  
    fazaSJ=4;  
    fazaZI=-4;  
}  
else if(faza1==3 && faza2==2){  
    fazaSJ=2;  
    fazaZI=-2;  
}  
else if(faza1==3 && faza2==3){  
    fazaSJ=0;  
    fazaZI=0;  
}
```

```
Serial.println(fazaSJ);
```

```
Serial.println(fazaZI);
```

```
// Postavljanje semafora u crveno u trajanju od 3 s.
```

```
red4(HIGH);
```

```
red3(HIGH);
```

```
red2(HIGH);
```

```

red1(HIGH);
delay(3000);
// Postavljanje žutog svjetla u stanje "1" na smjeru Z-I.
yellowX3(HIGH);
delay(2000);
//Smjer: Z-I.
// Ispitivanje aktivacije tipkala 1. Ako je aktivirano, pali zeleno svjetlo pješacima na tom
privozu.
if(tipkalo1)
  green2(HIGH);
//Palimo zeleno svjetlo na privozu Z-I.
green3(HIGH);
// Ispitivanje tipkala svakih 500ms za vrijeme trajanja prve 3 sekunde glavnog zelenog svjetla,
// u slučaju pritisnutog tipkala, pušta pješake.
for(i=0; i<20 + fazaZI; i++){
  if(tipkalo1 && i<6){
    Serial.println("Tipkalo 1 stisnuto u prve 3 sekunde");
    green2(HIGH);
    pj1=true;
  }
}
// Zatvaranje semafora i puštanje pješaka nakon 2s ako je pritisnuto tipkalo 2, prelazak na
fazu 2.
if(tipkalo2 && i<6){
  delay(2000);
  break;
}
else{
  delay(500);
}
}
// Gašenje zelenog svjetla u smjeru Z-I na način da prvo trepće zeleno pa se pali žuto i crveno
svjetlo.
blinkanjeZelenogZI(HIGH);
red2(HIGH);
if(pj1){
  tipkalo1=false;
  pj1=false;
}
delay(1000);
yellow3(HIGH);
delay(2000);
red3(HIGH);
delay(2000);
yellow1X(HIGH);

```

```

delay(2000);
// Smjer: S-J.
green1(HIGH);
// Ispitivanje aktivacije tipkala 2. Ako je, pali zeleno svijetlo pješacima na tom privozu.
if(tipkalo2)
  green4(HIGH);
// Ispitivanje tipkala svakih 500ms za vrijeme trajanja prve 3 sekunde glavnog zelenog svjetla.
// U slučaju pritisnutog tipkala, pušta pješake.
for(i=0; i<20 + fazaSJ; i++){
  if(tipkalo2 && i<6){
    green4(HIGH);
    pj2=true;
  }
  // Zatvaranje semafora i puštanje pješaka nakon 2s ako je pritisnuto tipkalo, prelazak na
  // fazu 2.
  if(tipkalo1 && i<6){
    delay(2000);
    break;
  }
  else{
    Serial.println(tipkalo1);
    delay(500);
    Serial.println(tipkalo1);
  }
}
// Gašenje trepćućeg zelenog svjetla u smjeru S-J i uključivanje crvenog svjetla za pješake.
blinkanjeZelenogSJ(HIGH);
red4(HIGH);
if(pj2){
  tipkalo2=false;
  pj2=false;
}
delay(1000);
yellow1(HIGH);
delay(2000);
}

```





Sveučilište u Zagrebu  
Fakultet prometnih znanosti  
10000 Zagreb  
Vukelićeva 4

## IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj \_\_\_\_\_ završni rad

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

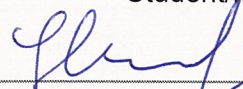
Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu \_\_\_\_\_ završnog rada

pod naslovom Mogućnosti primjene Arduino razvojne platforme za simulaciju rada raskrižja

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 10.9.2018

Student/ica:

  
\_\_\_\_\_  
(potpis)