Mogućnosti primjene Arduino razvojne platforme za simulaciju jednostavnog semaforiziranog pješačkog prijelaza

Vugrinec, Filip

Undergraduate thesis / Završni rad

2017

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:119:370771

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Filip Vugrinec

MOGUĆNOST PRIMJENE ARDUINO RAZVOJNE PLATFORME ZA SIMULACIJU JEDNOSTAVNOG SEMAFORIZIRANOG PJEŠAČKOG PRIJELAZA

ZAVRŠNI RAD

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
ODBOR ZA ZAVRŠNI RAD

Zagreb, 24. travnja 2017.

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

ZAVRŠNI ZADATAK br. 4108

Pristupnik: **Filip Vugrinec (0135227494)**
Studij: Promet
Smjer: Cestovni promet

Zadatak: **Mogućnosti primjene Arduino razvojne platforme za simulaciju jednostavnog semaforiziranog pješačkog prijelaza**

Opis zadatka:

Razvoj raznih rješenja sustava upravljanja semaforiziranim pješačkim prijelazima 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 jednostavni pješački prijelaz, te napisati potrebni programski kôd.

Zadatak uručen pristupniku: 28. travnja 2017.

Mentor:

doc. dr. sc. Edouard Ivanjko

Predsjednik povjerenstva za završni ispit:

Sveučilište u Zagrebu

Fakultet prometnih znanosti

ZAVRŠNI RAD

MOGUĆNOST PRIMJENE ARDUINO RAZVOJNE PLATFORME ZA SIMULACIJU JEDNOSTAVNOG SEMAFORIZIRANOG PJEŠAČKOG PRIJELAZA

POSSIBILITY OF APPLYING THE ARDUINO DEVELOPMENT PLATFORM FOR SIMULATION OF A SIMPLE PEDESTRIAN CROSSING TRAFFIC LIGHT

Mentor: doc. dr. sc. Edouard Ivanjko

Student: Filip Vugrinec

JMBAG: 0135227494

Zagreb, rujan 2017.

SAŽETAK

Ovaj završni rad obrađuje problematiku upravljanja jednostavnim semaforiziranim pješačkim prijelazom. U radu se opisuje simulacija jednog jednostavnog pješačkog prijelaza koristeći razvojnu platformu „Arduino“ koja predstavlja jednu od mnogih mogućih rješenja za simulaciju različitih sustava upravljanja u cestovnom prometu. Sastavljeno je i sklopovlje simulatora semafora za jednostavni pješački prijelaz te napisan pseudokôd i programski kôd, kojim se upravlja radom semafora.

NASLOV: Mogućnost primjene Arduino razvojne platforme za simulaciju jednostavnog semaforiziranog pješačkog prijelaza.

KLJUČNE RIJEČI: Arduino, ciklus, faza, interval, Matsonova metoda, Websterova metoda, semafor, pješački prijelaz.

SUMMARY

This thesis addresses the issue of control of a simple signalized pedestrian crossing. The simulation of a simple pedestrian crossing using the Arduino development platform, which is one of many possible solutions for simulation of various road traffic management systems, is described. The circuit for a simulator of a simple pedestrian crossing controlled with a traffic light is presented including the pseudocode and program, which manages the traffic lights.

TITLE: Possibility of Applying the Arduino Development Platform for Simulation of a Simple Pedestrian Crossing Traffic Light

KEYWORDS: Arduino, cycle, phase, interval, Matson method, Webster method, traffic light, pedestrian crossing

Sadržaj

1. Uvod 1
2. Problem upravljanja pješačkim prijelazom 2
2.1. Obilježja pješačkih prijelaza 2
2.2. Mjere za povećanje sigurnosti i mobilnosti pješaka 3
2.2.1. Najava pješaka 3
2.2.2. Pješački otoci 4
2.2.3. Podno osvjetljenje pješačkih prijelaza 4
2.2.4. LED prometna svijetla 5
2.2.5. Pješački prijelaz s odvojenim tokovima prometa 5
2.3. Određivanje signalnog plana 6
2.3.1. Metoda pokušaja i pogrešaka 6
2.3.2. Metoda po Matsonu 7
2.3.3. Metoda po Websteru 8
3. Razvojno okruženje Arduino 11
3.1. Arduino UNO 11
3.2. Arduino DUE 13
4. Sklop za upravljanje semaforom 14
5. Programska logika 16
5.1. Dijagram toka 16
5.2. Pseudokôd 18
6. Testiranje simulatora 20
7. Zaključak 21
Popis literature 22
Popis ilustracija 23
Popis formula 24
Popis kratica 25
Prilog 1: Programski kôd za upravljanje semaforom 26

1. Uvod

Pješačenje predstavlja najčešći način kretanja ljudi stoga pješaci, kako zbog zastupljenosti tako i zbog slabe zasićenosti, čine najviše ugroženu skupinu od svih sudionika u prometu. Iz tog razloga je potrebno provesti posebne mjere kako bi se smanjili rizici i broj stradalih pješaka u prometu. Povećanje brzine kretanja posljedica je suvremenog načina življenja, a prednosti pješačkog prometa leže u njegovoj neovisnosti o različitim oblicima prijevoza, velikim mogućnostima izbjegavanja prometnih zagušenja i povoljnim učincima na zdravlje.

Problem pješaka u cestovnom prometu, leži u lošem planiranju prometa i nepoštivanju propisa u prometu. Sigurnost pješačkog prijelaza očituje se kroz statističke podatke o broju prometnih nesreća, stradalih osoba i vremenskim utjecajima na sigurnost. Sam problem pješaka može se sažeti u sljedećem: *„Zabrinjavajuće je da godišnje u Europi strada približno 8.000 pješaka, od kojih svaki četvrti strada na pješačkom prijelazu. Testiranje je provedeno na 215 pješačkih prijelaza u 17 europskih glavnih gradova: Amsterdam, Barcelona, Berlin, Brussels, Kopenhagen, Helsinki, Ljubljana, London, Madrid, München, Oslo, Paris, Rim, Stockholm, Beč, Zagreb, Zurich. Kako bi procjenu mogli uspoređivati između različitih gradova, svi prijelazi su bili statistički odabrani unutar temeljito definiranih područja svakog grada. U Republici Hrvatskoj je testiran Zagreb, koji je dobio ukupnu ocjenu „Dobro“. Rezultat istraživanja pokazao je da pješački prijelazi zahtijevaju značajna poboljšanja s obzirom na sigurnost“* .[7]

Ovaj rad sadrži sedam poglavlja:

1. Uvod;
2. Problem upravljanja pješačkim prijelazom;
3. Razvojno okruženje Arduino;
4. Sklop za upravljanje semaforom;
5. Programska logika za upravljanje semaforom;
6. Testiranje simulatora;
7. Zaključak.

S obzirom na veliki broj stradalih pješaka, policija ističe da su najčešće pogreške vozača zbog kojih stradavaju pješaci neprilagođena ili nepropisna brzina kretanja vozila. Najčešće pogreške pješaka su nepropisan prijelaz preko kolnika i to nekorištenjem pješačkog prijelaza ili pješačkog pothodnika i prelaženjem na crveno svjetlo za pješake, dok ostali uzroci mogu biti loša vidljivost zbog nepovoljnih vremenskih uvjeta, smanjena preglednost na cesti, nepravilno označeni pješački prijelazi.

U ovom radu bit će obrađeno sklopovsko i programsko rješenje za simulaciju jednostavnog pješačkog prijelaza uporabom Arduino razvojne platforme.

2. Problem upravljanja pješačkim prijelazom

Koliko god se trudili povećati sigurnost pješačkih prijelaza, uvijek postoji prostor za poboljšanje. Već samim time što se i kultura sudionika u prometu konstantno mijenja, potrebno je proučavati promjene u ponašanju sudionika u prometu te ih obrazovati, a prometnu infrastrukturu prilagođavati.

2.1. Obilježja pješačkih prijelaza

Pješački prijelaz je posebno obilježena zona cestovne prometnice koju pješaci, nakon što napuste nogostup i stupe na kolnik („zebru“), koriste za prelazak na drugu stranu prometnice.
Prijelaz preko prometnice rješava se na nekoliko načina:

- obilježenim pješačkim prijelazom,
- semaforom,
- školskim patrolama,
- prometnom policijom.

Obilježavanjem pješačkog prijelaza na vidljiv i propisan način upozorava vozače na mogući nailazak pješaka. Pješački prijelaz postavlja se pod pravim kutom u odnosu na os prometnice kako bi pješaci prelazili cestu najkraćim putem, ili u smjeru kretanja pješačkog toka ako se ceste ne sijeku pod pravim kutom.

Osnovni cilj signalizacije (horizontalne i vertikalne) je pravovremeno obavještavanje svih sudionika u prometu o mogućim opasnostima. Svjetlosna signalizacija primjenjuje se u slučajevima velikog intenziteta prometa ili slabije sigurnosti prometa, te se iskazuje kao osobito korisna sigurnosna mjera za pješake.

„Zbog vlastite sigurnosti pješaci ne smiju:

- *prelaziti kolnik izvan obilježenih pješačkih prijelaza ako u naseljenom mjestu nisu udaljeni više od 50 metara, a izvan naselja više od 100 metara od pješačkog prijelaza,*
- *stupiti na kolnik i prelaziti preko kolnika bez prethodne provjere da to mogu učiniti na siguran način,*
- *prelaziti kolnik na znak crvenog svjetla na semaforima,*
- *pretrčavati kolnik ispred nailazećih vozila.*

Pješaci imaju zakonsku obvezu kretanja po nogostupu. No, kretanje pješaka u naseljima gdje nema javne rasvjete i izgrađenih nogostupa ili izvan naseljenih mjesta postiže se kretanjem uz lijevi rub kolnika u smjeru kretanja, a noću i danju u slučaju smanjene vidljivosti moraju biti osvijetljeni ili označeni reflektirajućim materijalom.“[5]

2.2. Mjere za povećanje sigurnosti i mobilnosti pješaka

Propisno postavljanje pješačkih prijelaza izravno utječe na razinu sigurnosti, neke od mjera za povećanje sigurnosti i mobilnosti pješačkih prijelaza su:

- najava pješaka;
- pješački otoci;
- podno osvjetljenje pješačkih prijelaza;
- LED prometna svijetla;
- odvajanje tokova pješačkog prometa.

2.2.1. Najava pješaka

Sklop za najavu pješaka sastoji se od tipkala koje omogućava postavljanje zahtjeva pješaka za zelenim svjetlom i davača zvučnog signala koji isprekidanim tonom označava stanje pješačkog zelenog svjetla, te na taj način omogućava lakše snalaženje slijepim i slabovidnim osobama u prometu.

Svaki semafor ima upravljački uređaj koji može biti programabilni logički kontroler (engl. Programmable Logic Controller, PLC). PLC ima mogućnost dogradnje dodatnih senzora za detekciju prisutnosti vozila (tzv. „induktivna petlja“). Na osnovu informacija primljenih od senzora prisutnosti vozila, pokreće se automatska promjena faza rada semaforskog uređaja prema prethodno unesenom kôdu.

2.2.2. Pješački otoci

Pješački otok je uzdignuta ili na drugi način obilježena površina koja se nalazi na kolniku i koja je određena za privremeno zadržavanje pješaka koji prelaze preko kolnika ili u vozilo i izlaze iz vozila javnog prijevoza [5]. Svrha otoka je zaštititi pješake prilikom prelaska na drugu stranu prometnice.



Slika 1: Pješački otok [10]

2.2.3. Podno osvjetljenje pješačkih prijelaza

U nastojanju da pješački prijelazi budu što vidljiviji noću, a da ujedno utječu na percepciju vozača IBM je razvio pješački prijelaz na kojem zebra svijetli noću. Takav pješački prijelaz počinje svijetliti noću kada se pješak nalazi pred zebrom. Osim vidljivosti takav pješački prijelaz signalizira vozaču namjeru pješaka da prijeđe prometnicu. Nakon što je pješak prešao na drugu stranu prometnice, zebra se gasi i prelazi u stanje mirovanja do ponovne aktivacije detektora koji registrira prisutnost pješaka.



Slika 2: Podno osvjetljenje pješačkog prijelaza [8]

2.2.4. LED prometna svijetla

Svrha i cilj LED svijetla je povećanje sigurnosti pješaka.
Prednosti navedene mjere su:

- povećanje vidljivosti pješačkog prijelaza;
- bolja uočljivost pješaka;
- pravovremeno obavještavanje vozača o nailasku na pješački prijelaz;
- stvaraju osjećaj sigurnosti za pješake.



Slika 3: LED prometna signalizacije pješačkog prijelaza [9]

2.2.5. Pješački prijelaz s odvojenim tokovima prometa

Brzina prelaska preko pješačkog prijelaza ovisi o broju pješaka koji prelazi prometnicu, a nailaskom većeg broja pješaka smanjuje se brzina prelaska. Uzrok toga je jednosmjeran tok pješačkog prometa uslijed kojeg dolazi do međusobne interakcije pješaka. Rješenje navedenog problema je u postavljanju pješačkog prijelaza s odvojenim tokovima pješačkog prometa. Usmjeravanje toka posebno je važno u većim urbanim sredinama kako bi se u što kraćem vremenu cestovna prometnica oslobodila od pješaka.



Slika 4: Pješački prijelaz s odvojenim tokovima [13]

2.3. Određivanje signalnog plana

Kod svjetlosnih uređaja za upravljanje motornim prometom tri su osnovna pojma:

- **Ciklus** je vrijeme od početka paljenja jedne kombinacije signalnih pojmova do ponovnog paljenja te iste kombinacije, odnosno vrijeme u kojem se izmjene sve faze u tom ciklusu. Trajanje ciklusa u dvofaznom sustavu upravljanja prometom je 40 – 60 s, kod trofaznog 60 – 90 s, a četverofaznog sustava 80 – 120 s.
- **Faza** je dio ciklusa u kojem jedna ili više skupina vozila ili pješaka ima slobodan prolaz. Trajanje pojedinih faza ne smije biti kraće od 15 sekundi.
- **Interval** je vrijeme trajanja bilo koje svjetlosne i/ili zvučne informacije.

Prema međunarodnom sustavu koji se primjenjuje i u Republici Hrvatskoj, redoslijed uključivanje signala (svjetala semafora) je (zeleno-žuto-crveno) i (crveno-crveno i žuto-zeleno). Crveno svijetlo ostaje upaljeno i za vrijeme žutog intervala prije promjene na zeleno svijetlo. Odnos između vremena trajanja zelenog intervala i ukupne duljine ciklusa posebno je važno za optimalnu propusnu moć.

Najvažnije metode za proračun dužine ciklusa i faza su:

- metoda pokušaja i pogrešaka,
- metoda po Matsonu,
- metoda po Websteru.

2.3.1. Metoda pokušaja i pogrešaka

Pri primjeni metode pokušaja i pogrešaka, potrebno je imati podatke o:

- prometnom opterećenju raskrižja;
- prometno opterećenje raskrižja u vršnom satu po minutama;
- vrijeme prosječnog intervala u slijedu vozila.

Određivanje duljine ciklusa obavlja se prema slijedećoj metodi:

Uz pretpostavku da je trajanje ciklusa **C_1 40 – 60 s** (iznimno 80 s), na temelju maksimalnog 15 minutnog vršnog opterećenja u satu izračuna se **koliko vozila maksimalno opterećenog privoza** pristiže na raskrižje u jednom ciklusu q_{c1} **voz/ciklusu.** Pomnožen sa **prosječnim intervalom slijeda h_1 (s)** određuje se **trajanje faze za promatrani prilaz f_1 (s)** prema formuli (1)**:**

$$f_I = q_{c1} \cdot h_1\ [s], \tag{1}$$

gdje je:

- f_I – faza $[s]$;
- q_{c1} –maksimalno opterećenje privoza na raskrižju [voz/ciklusu];
- h_1 – prosječni interval slijeda.

Ukoliko je trajanje faze za promatrani prilaz f_I (s) pola duljine ciklusa $(f_I < \frac{c_1}{2})$ postupak se ponavlja s kraćim ciklusom. Ukoliko je trajanje faze f_1 jednako ili veće od polovine duljine ciklusa $(f_I \geq \frac{c_1}{2})$ postupak se ponavlja i za slabije opterećene prilaze i određuje se **$faza_{II}$**. Ukoliko je $f_I + f_{II}$ < od C_1 ponavlja se isti postupak s kraćim ciklusom C_2 . Ukoliko je $f_I + f_{II}$ > od C_1 određuje se duljina ciklusa C_3 i postupak se ponavlja dok se ne odredi optimalna duljina trajanja ciklusa.

2.3.2. Metoda po Matsonu

Za primjenu ove metode potrebno je uzeti u obzir slijedeće čimbenike:

1. Čimbenik tromosti **K** pri prolasku prvog vozila koje započinje kretanje;
2. Vremenski interval **H**, razmak pristizanja vozila na raskrižje koji je zadan izrazom (2):

$$H = \frac{Q}{3600}\,[\mathrm{s}], \tag{2}$$

   gdje je:

   - H – Vremenski interval $[s]$;
   - Q – Prometno opterećenje (voz/h).
3. Vremenski interval **D** za koji vozila napuštaju raskrižje, mjeren od ulaska vozila u raskrižje;
4. Broj faza **n** u ciklusu;

Duljina ciklusa određuje se prema izrazu (3):

$$C = \frac{\sum_{x=1}^{n} K_x}{1-\sum_{x=1}^{n} H_x \cdot D_x}\,[s], \qquad (3)$$

gdje je:

- C - duljina ciklusa $[s]$;
- K_x - čimbenik tromosti;
- H_x - vremenski interval, razmak pristizanja vozila na križanje;
- D_x - vremenski interval za koji vozilo napušta križanje
- n – broj faza u ciklusu.

Uz opterećenje izraženo u putničkim automobilima i prihvaćanjem idealnih performansi vozila srednje klase, vrijednost čimbenika tromosti iznosi K = 4,75, vrijeme intervala napuštanja raskrižja D = 2,1 i broj faza u ciklusu n = 2, duljina ciklusa tada iznosi prema izrazu (4):

$$C = \frac{2 \cdot 4{,}75 \cdot 3600}{3600-2{,}1 \cdot (Q_1+Q_2)}\,[s], \qquad (4)$$

gdje je:

- C duljina ciklusa $[s]$;
- Q_1 veličina prometa na pravcu 1 [voz/s];
- Q_2 veličina prometa na pravcu 2 [voz/s].

Nedostatak Matsonove metode je što ne uzima u obzir utjecaj teretnih vozila i vozila koja skreću u lijevo.

2.3.3. Metoda po Websteru

Najviše se primjenjuje za proračun rada semafora na pojedinačnim raskrižjima. Prilikom određivanja duljina ciklusa i faza semaforiziranog raskrižja potrebno je uzeti u obzir slijedeće elemente:

- Količina prometa S_i pri kojoj nastaje zasićen tok na prilazu [voz/h] – (zasićena količina toka je broj vozila na sat koje može prihvatiti grupa trakova, pretpostavkom da je zelena faza uvijek na raspolaganju na raskrižju, odnosno odnos zelenog g/C je 1,00);
- Stvarna količina prometa q_i iz promatranog prilaza [voz/h];
- Koeficijent iskorištenja prilaza $y_i = \frac{q_i}{S_i}$.

Duljina ciklusa C u tom slučaju može se izračunati prema jednadžbi (5):

$$C = \frac{1{,}5 \cdot L + 5{,}0}{1.0 - y}[s], \tag{5}$$

gdje je:

- C – duljina ciklusa $[s]$;
- L - ukupno izgubljeno vrijeme na raskrižju u jednom ciklusu [s];
- y - $\sum_{i=1}^{n} y_1$ koeficijent iskorištenja raskrižja.

Duljina trajanja f_i može se izračunati prema izrazu (6):

$$f_1 = (C - L) \cdot \frac{y_i}{y}\ [s], \tag{6}$$

gdje je:

- f_1- duljina trajanja $[s]$;
- C – duljina ciklusa $[s]$;
- L - ukupno izgubljeno vrijeme na raskrižju u jednom ciklusu [s];
- y_i- koeficijent iskorištenja prilaza;
- y - koeficijent iskorištenja križanja.

Prema općoj metodi, kod ciklusa s dvije faze, razdioba na faze može se izračunati prema slijedećim izrazima (7) i (8):

$$f_A = \frac{C_0 \cdot N_A}{N_A + N_B}, \tag{7}$$

$$f_B = \frac{C_0 \cdot N_B}{N_A + N_B}, \tag{8}$$

gdje su:

- f_A i f_B – duljine trajanja zelenog svjetla za prilaz A i B izraženo u [s];
- N_A i N_B opterećenje PAJ/h iz prilaza A i B;
- C_0 – ukupno zeleno svijetlo za vrijeme jednog ciklusa u [s].

Trajanje ciklusa ovisi od broja i dužine signalnih pojmova, kao broj faza. Pokazatelj C_0 je dominantan element proračuna kapaciteta razina usluge površinskog raskrižja sa svjetlosnom signalizacijom. Efektivnost odvijanja prometa na raskrižju sa svjetlosnom signalizacijom ovisi od dužine i učestalosti pojave zelenog signalnog pojma. Oba zahtjeva ne mogu se istovremeno zadovoljiti jer dužina trajanja zelenog vremena za jedan smjer uzrokuje duže trajanje crvenog vremena za suprotni smjer.

Vremenski gubitci, čekanje i skupljanje vozila na semaforiziranim raskrižjima, usko su povezani s pojavom sve većeg broja vozila na mreži prometnica. Na semaforiziranim raskrižjima vozila moraju dijeliti prednost prolaska s nekim drugim konfliktnim tokom, a takav sustav stvara vremenske gubitke vozila.

Vremenski gubici se definiraju kao razlika između stvarno utrošenog vremena za prolaz kroz semaforizirano raskrižje i vremena koje bi bilo potrebno za prolaz kada bi promatrani prilaz imao stalno zeleni signalni pojam. Tako definirani vremenski gubici predstavljaju jedan od osnovnih kriterija za određivanje dužine ciklusa i preraspodjele zelenog vremena unutar ciklusa na semaforiziranim raskrižjima.

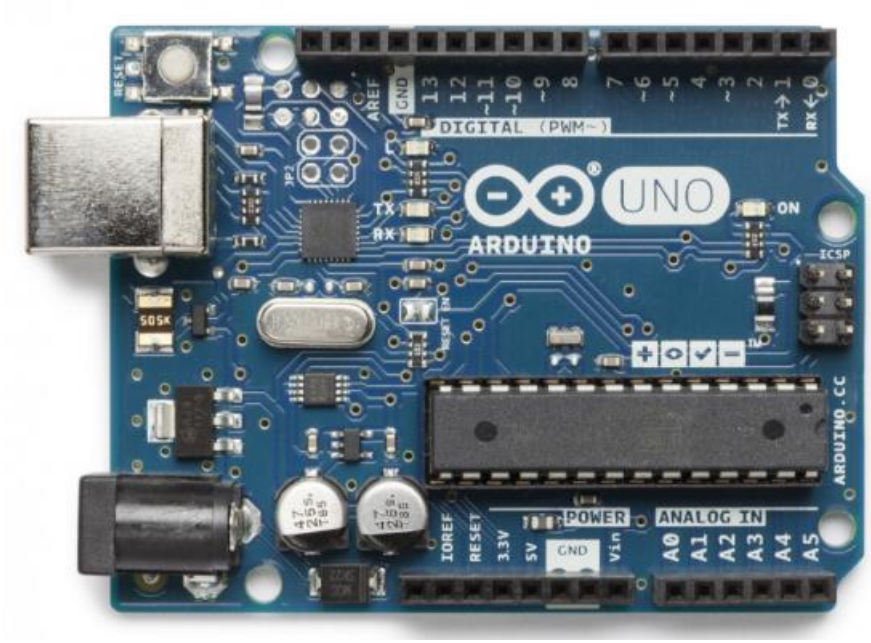
3. Razvojno okruženje Arduino

Arduino platforma (pločica) sastoji se od:

- mikroprocesora;
- linearnog naponskog regulatora od 5V;
- USB utora koji omogućava povezivanje pločice s računalom;
- ulazno izlaznih pinova;
- LED-a (signalizacija napajanja, testa, prijenos podataka prema i od nadzornog računala i dr.).

3.1. Arduino UNO

Arduino UNO je mikrokontroler s 14 ulaza i izlaza, 6 analognih ulaza, USB priključkom, utičnicom, ICSP (engl. In Circuit Serial Programming) i gumb za resetiranje. Jednostavno ga je spojiti na računalo pomoću USB kabela.



Slika 5: Arduino UNO razvojna pločica

Za programiranje Arduina se koristi Arduino integrirano razvojno okruženje (engl. Integrated Development Environment, IDE) koje je zasnovano na Processingu, odnosno besplatnoj programskoj podršci (engl. Freeware Software) kojim se programira Arduino sklopovlje na Wiring programskom jeziku (sličan C/C++ jeziku). Integrirano razvojno okruženje Arduina je aplikacija napisana u Java jeziku. Sastoji se od editor-a programskog kôda.

Processing je programski jezik i integrirano razvojno okruženje otvorenog kôda (engl. Open Code). Moguće je generiranje zvuka, slanje i primanje podataka u raznim formatima, kao i slanja 2D i 3D datoteka.

Budući da se Arduino programi pišu u C/C++ programskom jeziku, korisnici moraju definirati samo dvije funkcije kako bi napravili izvršni program. Te funkcije su:

- **setup()** – funkcija koja se izvršava jednom na početku i služi za početna podešavanja;
- **loop()** – funkcija koja se izvršava u petlji cijelo vrijeme dok se ne isključi napajanje pločice.

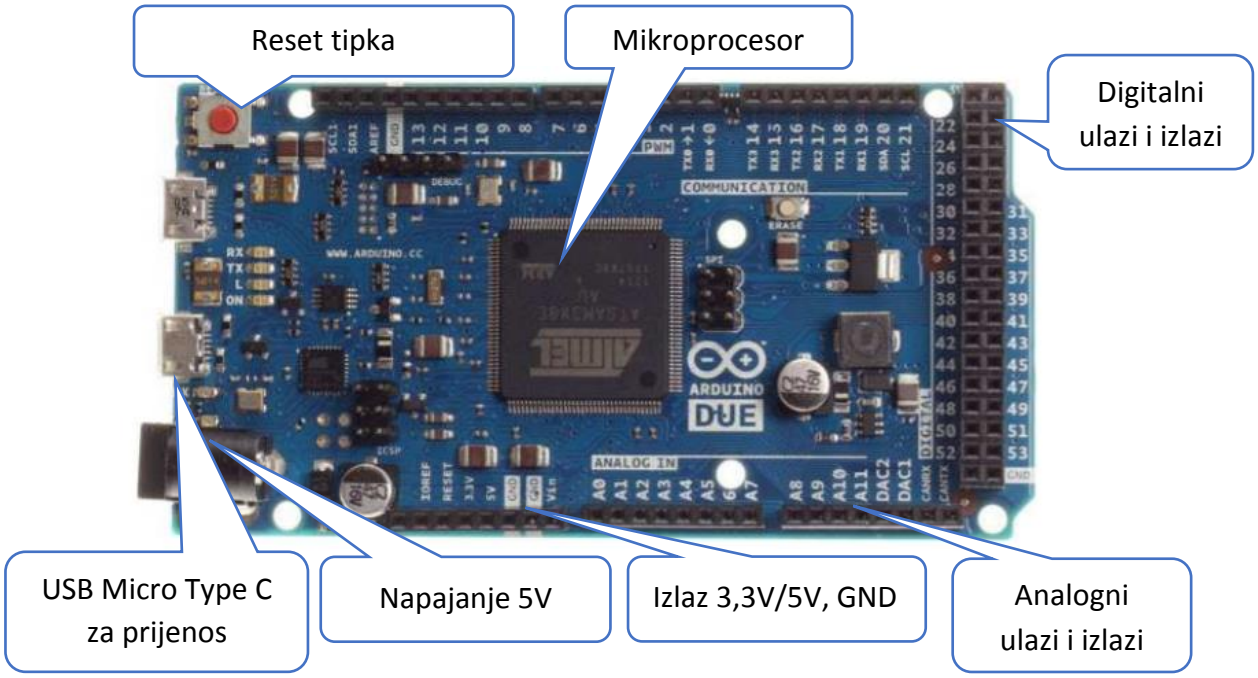
Wiring je platforma za razvoj elektronskih prototipova, a čine je:

- programski jezik,
- integrirano razvojno sučelje i
- mikrokontrolerska pločica.

Arduino okruženje dozvoljava razna proširenja od već gotovih komponenti do specijaliziranih komponenti koje je izradio programer ili korisnik.

3.2. Arduino DUE

Arduino DUE je prva Arduino pločica koja se zasniva na 32-bitnoj ARM jezgri, sadrži 54 digitalnih ulaza i izlaza, s 12 analognih ulaza, savršena je ploča za složenije projekte većih razmjera, te je korištena u ovom završnom radu.



Slika 6: Arduino DUE pločica

	ARDUINO UNO	ARDUINO DUE
Analogni ulaz	6, 10-bitni	12, 10-bitni
Analogni izlaz	6, PWM, 8-bitni	2, DAC, 10-bitni
Digitalni ulaz/izlaz	14, 6 PWM, 8-bitni	54, 12 PWM 8-bitni
Mikro kontroler	ATMega 328	AT91SAM3X8E
Radni takt	16 MHz	84 MHz
Flash memorija	32 KB	512 KB
Sram	2 KB	96 KB

Slika 7: Tehničke karakteristike UNO i DUE pločice

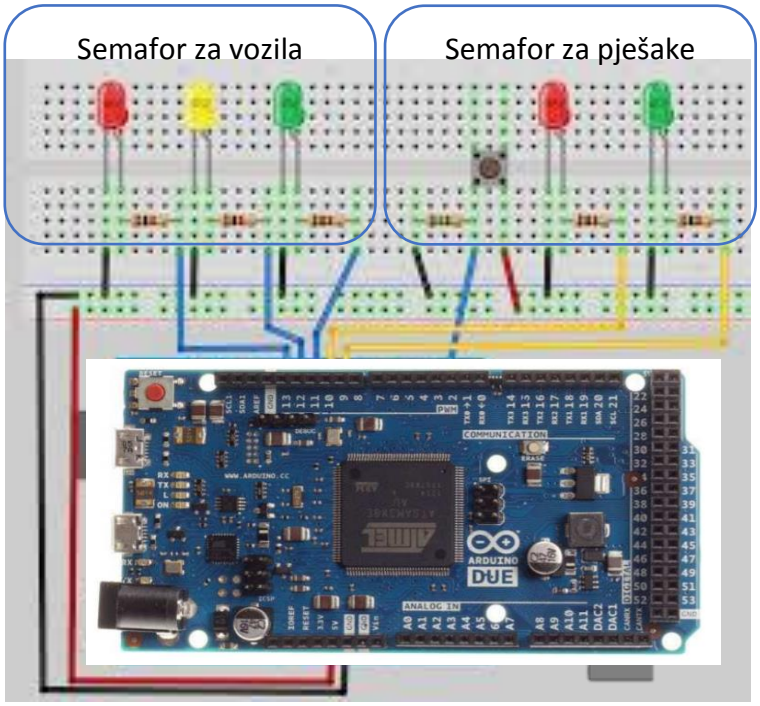
4. Sklop za upravljanje semaforom

Na slici 8 prikazana je shema spajanja sklopa s jednim semaforom za vozila i jednim semaforom za pješake. Na samom sklopu paralelno će se spojiti dva semafora za vozila i dva za pješake budući se uzela za primjer prometnica s dvosmjernim prometom.

Nakon što se prenese programski kôd u mikroprocesor kontrolera dovoljno je samo napajanje od 5V istosmjerno kako bi sklop radio. Više nije potrebno spajanje na računalo, izuzev ako je potrebno izvršiti korekciju kôda i praćenje trenutnog stanja semafora na računalu.

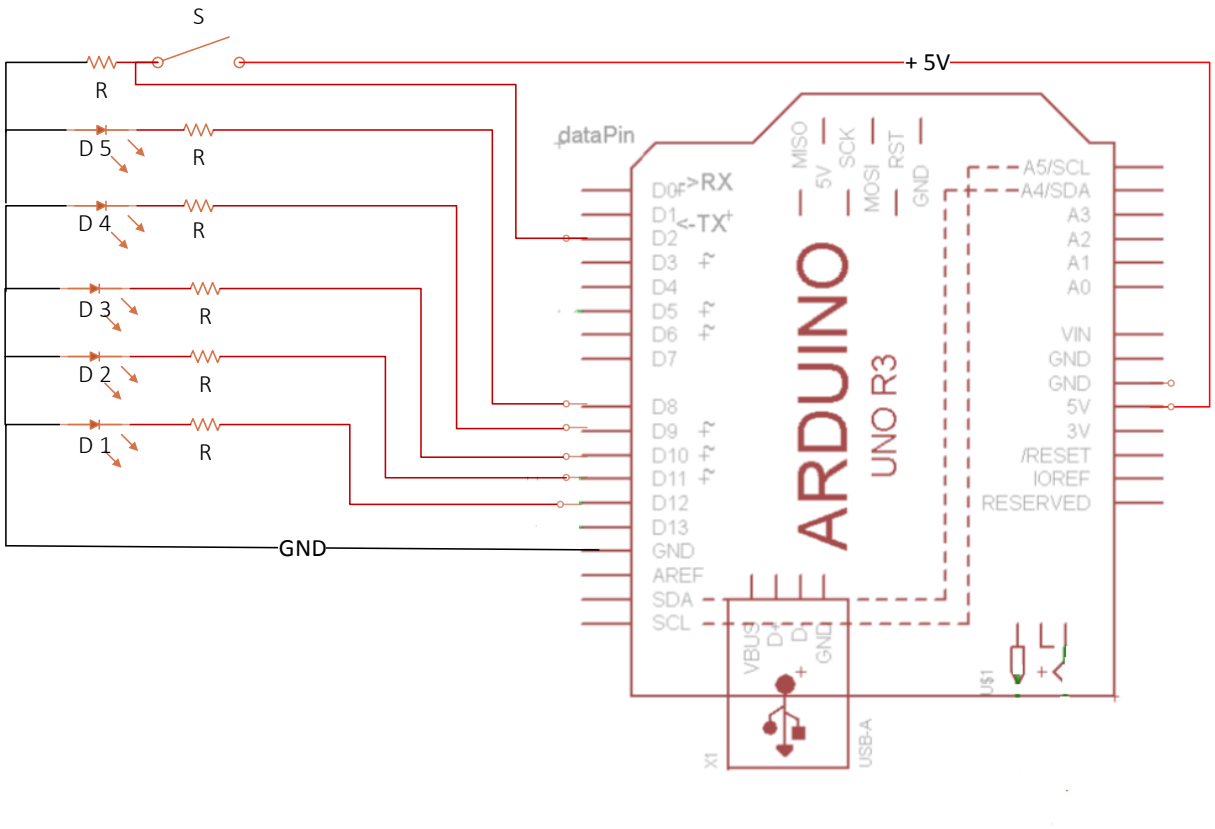
Tipkalo simulira zahtjev pješaka. LED diode su opremljene pred otporom R = 220 kΩ radi zaštite od prevelike struje.

Za izradu prototipa, koristi se univerzalna pločica na koju su postavljeni elementi sklopa koji se mikro kablom povezuje s ulazima i izlazima Arduino pločice. Na univerzalnu pločicu dovodi se napajanje od +5V i GND.



Slika 8: Prikaz spajanja na univerzalnoj pločici za brzu izradu prototipa

Na slici 8 s lijeve strane nalazi se sklop koji simulira jedan semafor za vozila, dok se na maketi paralelno spajaju dva semafora za vozila. S desne strane nalazi se sklop koji simulira semafor za pješake, dok se na maketi paralelno spajaju dva semafora za pješake.



Slika 7: Električna shema spajanja

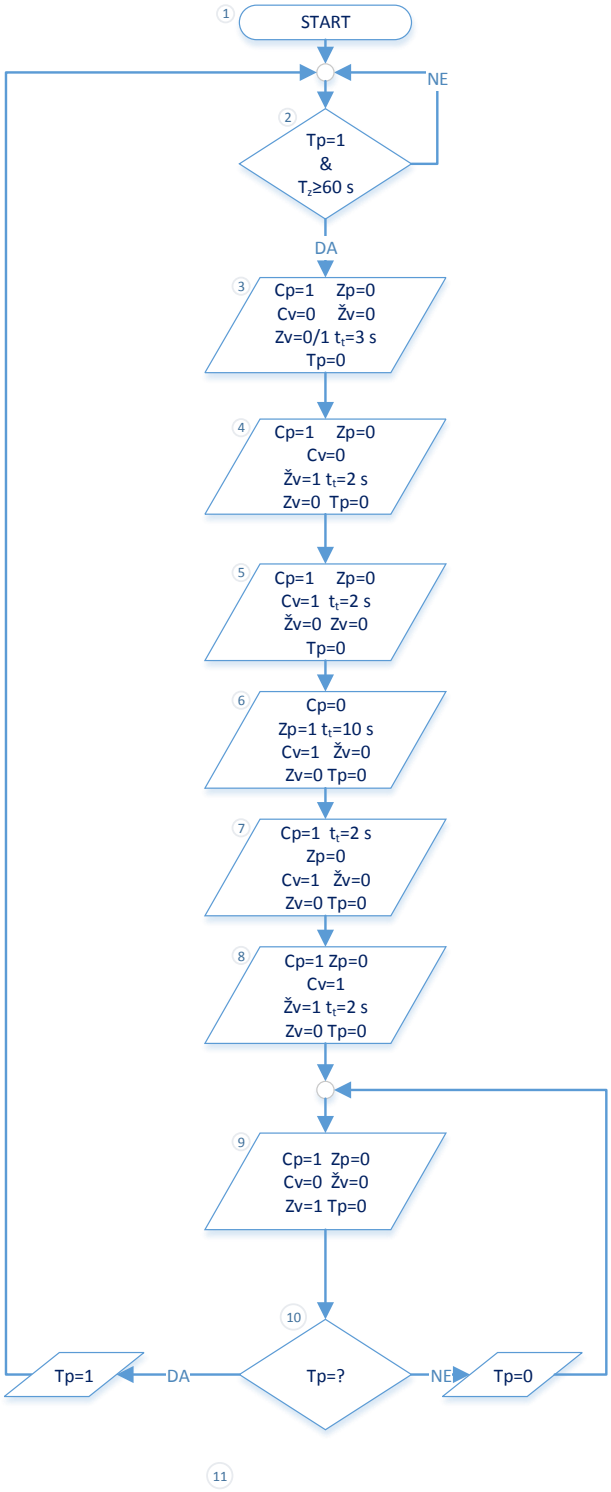
D1, D2, D3 su LED diode semafora za vozila, dok su D3 i D4 LED diode semafora za pješake. S je tipkalo za pješake. Svaki element ima pred otpor od 220 OHM. Sklop se napaja istosmjernom strujom od +5V preko USB priključka s računala ili vanjske baterije.

Pritiskom na sklopku S signal se šalje preko kontakta D2 na pločicu gdje kontrolera pokreće petlju koja je prethodno programirana i učitana u EPROM.

5. Programska logika

U ovom poglavlju prikazati će se logika kôda kroz dijagram toka, pseudokôd, dok je programski kôd prikazan u Prilogu 2 ovog završnog rada.

5.1. Dijagram toka



Slika 8: Dijagram toka rada semafora

Opis izvršavanja dijagrama toka:

1. Uključivanje napajanja

2. Ako je tipkalo za pješake pritisnuto, a prošlo je 60 s od zadnjeg pritiska, pokreće se proces prebacivanja faza. Ako uvjet nije ispunjen ne mijenja se ništa.

3. Ako je uvjet ispunjen, zelen svjetlo za vozila počinje treperiti u trajanju od 3 sekunde.

4. Gasi se zeleno svjetlo za vozila i pali žuto svjetlo u trajanju od 2 sekunde.

5. Gasi se žuto svijetlo i pali crveno za vozila uz sigurnosnu zadršku od 2 sekunde.

6. Gasi se crveno svjetlo za pješake i pali zeleno za pješake u trajanju od 10 sekundi.

7. Po isteku vremena gasi se zeleno za pješake i pali crveno za pješake. Crveno za vozila je upaljeno.

8. Crveno za pješake i za vozila su upaljena. Pali se žuto za vozila u trajanju od 2 sekunde.

9. Crveno za pješake je upaljeno. Crveno i žuti za vozila se gase, a pali se zeleno za vozila.

10. Ako se pojavi uvjet pritisnutog tipkala s preduvjetom da je od zadnjeg pritiska prošlo 60 sekundi, ponavlja se proces prebacivanja faza. U suprotnom je pješacima crveno a vozačima zeleno do ispunjavanja navedena dva uvjeta.

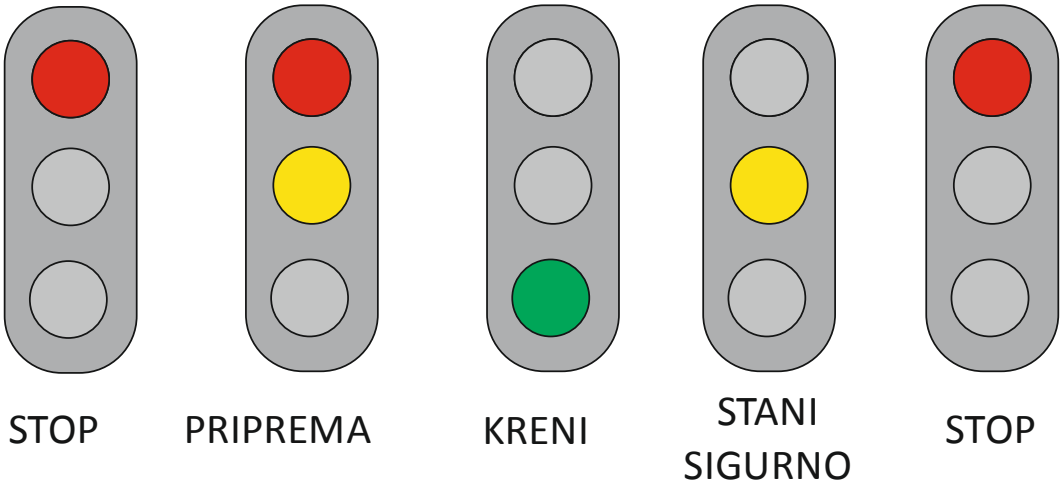
5.2. Pseudokôd

```
TURN Cp ON                                              (upali crveno za pješake)
TURN Zv ON                                              (upali zeleno za vozače)
BEGIN LOOP                                              (započni petlju)
    IF BUTON Tp IS PRESSED AND  tz ≥ 60 SECONDS THEN    (ako je pritisnuto tipkalo i
                                                        prošlo je 60 sekundi od
                                                        zadnjeg pritiska onda)
    STOP WAITING                                        (prestani čekanje)
    END IF                                              (kraj IF naredbe)
    WAIT 2 SECONDS                                      (čekaj dvije sekunde)
    FLASH Zv FOR 3 SECONDS                              (treperi tri sekunde)
    TURN Zv OFF                                         (ugasi zeleno za vozače)
    TURN Žv ON                                          (upali žuto za vozače)
    WAIT 2 SECONDS                                      (čekaj dvije sekunde)
    TURN Žv OFF                                         (ugasi žuto za vozače)
    TURN Cv ON                                          (upali crveno za vozače)
    WAIT 2 SECONDS                                      (čekaj dvije sekunde)
    TURN Zp ON                                          (upali zeleno za pješake)
    WAIT 10 SECONDS                                     (čekaj deset sekundi)
    TURN Zp OFF                                         (ugasi zeleno za pješake)
    TURN Cp ON                                          (upali crveno za pješake)
    WAIT 2 SECONDS                                      (čekaj dvije sekunde)
    TURN Žv ON                                          (upali žuto za vozače)
    WAIT 2 SECONDS                                      (čekaj dvije sekunde)
    TURN Žv OFF                                         (ugasi žuto za vozače)
    TURN Cv OFF                                         (ugasi crveno za vozače)
    TURN Zv ON                                          (upali zeleno za vozače)

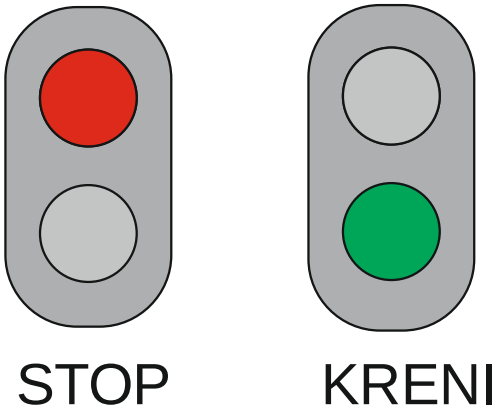
END LOOP
```

Opis pseudokôda

U postavkama je određeno da se kod uspostave naponskog stanja, upali crveni signalni pojam za pješake i zeleni za vozila. Prvih 60 sekundi ostati će takvo stanje bez mogućnosti za promjenom uslijed signala tipkala za pješake. Nakon što istekne navedeno vrijeme zatezanja, ukoliko se pritisne tipkalo nakon dvije sekunde će krenuti promjena faza iz zelenog signalnog pojma za vozače u crveni signalni pojam za vozače. Promjena se odvija na način da zeleni signalni pojam treperi tri sekunde nakon čega se gasi, a slijedi žuti signalni pojam od dvije sekunde. Po isteku dvije sekunde, gasi se žuti signalni pojam, a pali crveni signalni pojam. Nakon slijedeće dvije sekunde, pali se zeleni signalni pojam za pješake koji traje deset sekundi.



Slika 9: Redoslijed paljenja semafora za vozače s tumačenjem pojedinih faza



Slika 10: Redoslijed paljenja semafora za pješake s tumačenjem pojedinih faza

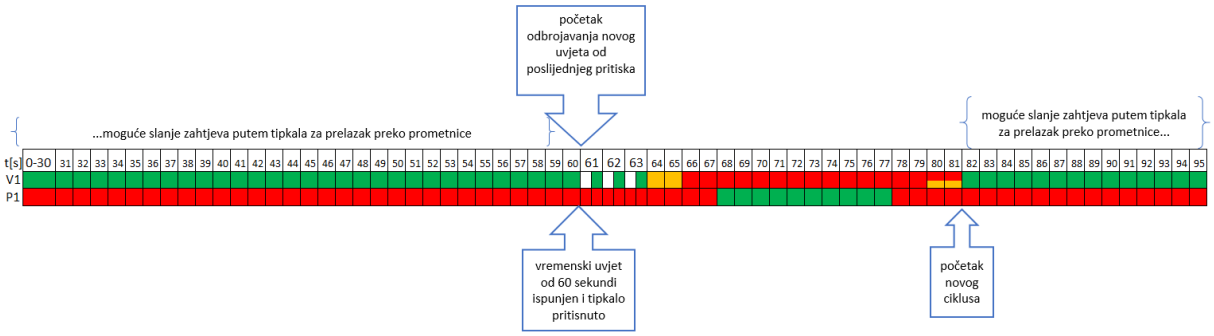
Sukladno Zakonu o sigurnosti prometa na cestama izrađen je shematski prikaz faza rada semafora sa značenjem pojedine faze.

6. Testiranje simulatora

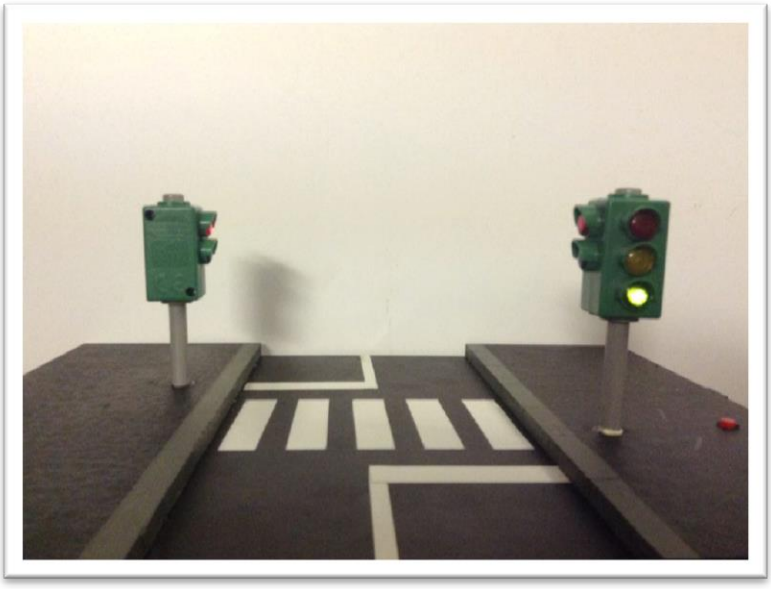
Najjednostavnije je signalni plan prikazati grafički na vremenskoj crti kako bi se dobio uvid u preklapanja signala i dužinu trajanja pojedinih faza. Na slici 10, prikazano je vremensko trajanje pojedinih faza. Grafički prikaz prikazuje situaciju rada semafora s nalogom pješaka.

Ukoliko pješak ne pritisne tipkalo i ne uputi nalog sučelju da uključi zeleno svijetlo pješacima, neće se uključiti signal za pješake već će cijelo vrijeme biti uključeno zeleno svjetlo radi propuštanja vozila. V1 je oznaka faze za vozila, dok je P1 oznaka faze za pješake.

Ukoliko pješak pritisne tipkalo, a ispunjen je uvjet da je prošlo 60 sekundi od zadnjeg pritiska tipkala, počinje prebacivanje iz zelene faze za vozila u crvenu fazu za vozila, a u zelenu fazu za pješake.



Slika 11: Signalni plan i linijski dijagram



Slika 12: Prikaz makete simulatora

7. Zaključak

Rješavanje pitanja sigurnosti svih sudionika u prometu je kontinuiran proces koji je moguće stalno poboljšavati. Razvidno je da neko od do sada usvojenih rješenja nije moguće izdvojiti kao najbolje što otvara prostor za daljnja poboljšanja. U radu je opisano rješenje za simulaciju upravljanja jednostavnim pješačkim prijelazom zasnovano na okruženju Arduino. Bez obzira što se ovdje radi o jednostavnijem sučelju s mogućnošću slobodnog programiranja moguće je na istoj platformi implementirati razna poboljšanja kao što su trepteća LED svijetla na kolniku prije pješačkog prijelaza te zvučnu signalizaciju za pješake.

Danas se u većini slučajeva koriste mikroprocesori s tranzistorom kao sklopka jer nema pokretnih dijelova te veće izdržljivosti i dugotrajnosti, no ugradnjom LED rasvjete umjesto klasične svjetiljke sa žarnom niti, počinje se primjenjivati upravljanje kontrolerima. Time se otvaraju nove mogućnosti nadogradnji i poboljšanja bez nužne potrebe za zamjenom postojećeg sklopovlja, već je sve promjene moguće napraviti programski, ponekad i daljinski iz središnjeg prometnog upravljačkog centra. Dodatno se time smanjuje potreba interveniranja na terenu te se otvara potreba za novim kadrovima (automatičarima, elektroničarima, informatičarima, naročito prometnim stručnjacima u prometnom upravljačkom centru), a ujedno i smanjuje potreba za klasičnim elektroinstalaterima.

Popis literature

[1.] Cerovac, V.: Tehnika i sigurnost prometa, Fakultet prometnih znanosti , Zagreb, 2001.

[2.] Legac, I. i koautori: Gradske prometnice, Fakultet prometnih znanosti , Zagreb, 2011.

[3.] Luburić, G.: Sigurnost cestovnog prometa 1, Nastavni materijal, Fakultet prometnih znanosti, Zagreb, 2010.

[4.] Urbanik, T., Tanaka, A.: Signal Timing Manual, US Department of Transportation, Washington D.C., 2008.

[5.] Zakon o sigurnosti prometa na cestama (N.N. 67/08, 48/10, 74/11, 80/13, 158/13, 92/14, 64/15)

[6.] https://www.arduino.cc/ (31.kolovoz 2017.)

[7.] http://www.hak.hr/sigurnost-u-prometu/projekti/eurotest/ (15.srpanj 2017.)

[8.] http://bit.ly/2wpytPS (15.srpanj 2017.)

[9.] http://bit.ly/2wtWO5B (15.srpanj 2017.)

[10.] http://bit.ly/2gvA01e (15.srpanj 2017.)

[11.] http://www.processing.org/ (17.srpanj 2017.)

[12.] http://wiring.org.co/ (17.srpanj 2017.)

[13.] http://bit.ly/2eMeaTM (17.srpanj 2017.)

Popis ilustracija

Slika 1: Pješački otok [10] 4
Slika 2: Podno osvjetljenje pješačkog prijelaza [8] 4
Slika 3: LED prometna signalizacije pješačkog prijelaza [9] 5
Slika 4: Pješački prijelaz s odvojenim tokovima [13] 5
Slika 5: Arduino UNO razvojna pločica 11
Slika 6: Arduino DUE pločica 13
Slika 7: Električna shema spajanja 15
Slika 8: Dijagram toka rada semafora 16
Slika 9: Redoslijed paljenja semafora za vozače s tumačenjem pojedinih faza 19
Slika 10: Redoslijed paljenja semafora za pješake s tumačenjem pojedinih faza 19
Slika 11: Signalni plan i linijski dijagram 20
Slika 12: Prikaz makete simulatora 20

Popis formula

(1) Faza
(2) Vremenski interval
(3) Duljina ciklusa
(4) Duljina ciklusa
(5) Duljina ciklusa
(6) Duljina trajanja faze
(7) Duljine trajanja zelenog svjetla za prilaz A
(8) Duljine trajanja zelenog svjetla za prilaz B

Popis kratica

Cp	crvena faza za pješake
Cv	crvena faza za vozila
Zp	zelena faza za pješake
Zv	zelena faza za vozila
Žv	žuta faza za vozila
Tp	Tipkalo za pješake
LED	(Light Emiting Diode) svjetleća dioda
PLC	(Programable Logical Controler) programibilni logički kontroler
IBM	(International Business Machines Corporation) međunarodna informatička tvrtka

Prilog 1: Programski kôd za upravljanje semaforom

```
// inicijalizacija varijabli
// pinovi
int btn = 2;
int v_red = 5;
int v_yellow = 4;
int v_green = 3;
int p_red = 12;
int p_green = 11;
boolean turn_p_green = false;
// inicijalizacija vremenskih vrijednosti u milisekundama
int blink_delay = 500;
int semaphore_switch_delay = 2000;
int light_switch_delay = 2000;
int people_semaphore_green_light_time = 10000;
int min_time_before_p_green = 60000;

int last_p_green = 0;
 // defaultna setup funkcija arduina koja se izvrši pri pokretanju
void setup() {
  // otvaranje serijskog porta za komunikaciju sa računalom
  Serial.begin(9600);
  // setiranje moda pojedinih pinova
  pinMode(p_green, OUTPUT);
  pinMode(p_red, OUTPUT);
  pinMode(v_green, OUTPUT);
  pinMode(v_yellow, OUTPUT);
  pinMode(v_red, OUTPUT);
  pinMode(btn, INPUT);
```

```
  // početno stanje semafora (za aute zeleno, pješake crveno)
  digitalWrite(v_green, HIGH);
  digitalWrite(p_red, HIGH);
}

// funkcija za prebacivanje semafora za aute na crveno
void switchToRedV() {
    blinkGreenV();
    Serial.print("VOZAČI: Gasim zeleno \n");
    digitalWrite(v_green, LOW);
    Serial.print("VOZAČI: Palim žuto \n");
    digitalWrite(v_yellow, HIGH);
    delay(light_switch_delay);
    Serial.print("VOZAČI: Gasim žuto \n");
    digitalWrite(v_yellow, LOW);
    Serial.print("VOZAČI: Palim crveno \n");
    digitalWrite(v_red, HIGH);
}

 // funkcija za prebacivanje semafora za aute na zeleno
void switchToGreenV() {
    Serial.print("VOZAČI: Palim žuto \n");
    digitalWrite(v_yellow, HIGH);
    delay(light_switch_delay);
    Serial.print("VOZAČI: Gasim crveno \n");
    digitalWrite(v_red, LOW);
    Serial.print("VOZAČI: Gasim žuto \n");
    digitalWrite(v_yellow, LOW);
    Serial.print("VOZAČI: Palim zeleno \n");
```

```
    digitalWrite(v_green, HIGH);
}
 // funkcija za prebacivanje semafora za pješake na crveno
void switchToRedP() {
    Serial.print("PJEŠACI: Gasim zeleno \n");
    digitalWrite(p_green, LOW);
    Serial.print("PJEŠACI: Palim crveno \n");
    digitalWrite(p_red, HIGH);
}
 // funkcija za prebacivanje semafora za pješake na zeleno
void switchToGreenP() {
    Serial.print("PJEŠACI: Gasim crveno \n");
    digitalWrite(p_red, LOW);
    Serial.print("PJEŠACI: Palim zeleno \n");
    digitalWrite(p_green, HIGH);
}
// funkcija za blinkanje zelenog svjetla na semaforu za aute tri puta
void blinkGreenV() {
    for (int i = 0; i < 3; i++) {
      digitalWrite(v_green, LOW);
      delay(blink_delay);
      digitalWrite(v_green, HIGH);
      delay(blink_delay);
    }
}
// funkcija koja poziva pomoćne funkcije i prebacuje semafor za pješake na zeleno,
// a semafor za aute na crveno
void pButtonPressed() {
    switchToRedV();
```

```
  delay(semaphore_switch_delay);
  switchToGreenP();
  // zabilježi broj milisekundi zadnjeg zelenog pješacima
  last_p_green = millis();
  delay(people_semaphore_green_light_time);
  switchToRedP();
  delay(semaphore_switch_delay);
  switchToGreenV();
}

void loop() {
  // učitaj stanje gumba
  int buttonState = digitalRead(btn);
  // ako je stanje HIGH tj gumb je stisnut postavi flag za paljenje semafora na true
  if (buttonState) turn_p_green = true;
  // ako je vrijeme zadnjeg zelenog pješacima manje od minimalnog vremena
  // koje treba proći prije nego pješacima se pali zeleno I ako je gumb stisnut
  // prebaci svjetla na semaforima
  if (turn_p_green && millis() - last_p_green >= min_time_before_p_green){
    // resetiranje flag-a
    turn_p_green = false;
    delay(2000);
    pButtonPressed();
  }
}
```

Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem da je **ZAVRŠNI RAD**
(vrsta rada)

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korištene bilješke i bibliografija. Izjavljujem da nijedan dio
rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da 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/diplomskog rada pod naslovom **MOGUĆNOST PRIMJENE ARDUINO RAZVOJNE PLATFORME ZA SIMULACIJU JEDNOSTAVNOG SEMAFORIZIRANOG PJEŠAČKOG PRIJELAZA**, na mrežnim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 28.08.2017.

Student/ica:
Vignjevac
(potpis)