

# Sklopovska i programska izrada prototipa autonomnog vozila zasnovanog na mobilnoj platformi i postupcima strojnog učenja

---

Krčmar, Tea

Undergraduate thesis / Završni rad

2022

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:200:351432>

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

*Download date / Datum preuzimanja:* **2024-12-13**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij računarstvo**

**SKLOPOVSKA I PROGRAMSKA IZRADA  
PROTOTIPA AUTONOMNOG VOZILA  
ZASNOVANOG NA MOBILNOJ PLATFORMI I  
POSTUPCIMA STROJNOG UČENJA**

**Završni rad**

**Tea Krčmar**

**Osijek, 2021.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 17.09.2021.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Tea Krčmar
<b>Studij, smjer:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R4081, 28.07.2017.
<b>OIB Pristupnika:</b>	99352098246
<b>Mentor:</b>	Prof.dr.sc. Goran Martinović
<b>Sumentor:</b>	Dino Kurtagić,
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Sklopovska i programska izrada prototipa autonomnog vozila zasnovanog na mobilnoj platformi i postupcima strojnog učenja
<b>Znanstvena grana rada:</b>	<b>Arhitektura računalnih sustava (zn. polje računarstvo)</b>
<b>Zadatak završnog rada:</b>	U teorijskom dijelu završnog rada potrebno je opisati i analizirati sklopovske i programske zahtjeve kojima se može omogućiti autonomna vožnja prototipa vozila. Nadalje, treba predložiti sklopovsko rješenje takvog sustava zasnovano na jednodimenzionalnom računaru i/ili mikroupravljaču (Raspberry Pi, Arduino), te odgovarajućim kamerama i senzorima. Također, treba osmisliti programsko rješenje koje omogućuje prikupljanje podataka s kamere i senzora, njihovu obradu, zaslone i prikazivanje rezultata.
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 1 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	17.09.2021.
<b>Datum potvrde ocjene od strane Odbora:</b>	22.09.2021.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 27.09.2022.

**Ime i prezime studenta:**

Tea Krčmar

**Studij:**

Preddiplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

R4081, 28.07.2017.

**Turnitin podudaranje [%]:**

10

Ovom izjavom izjavljujem da je rad pod nazivom: **Sklopovska i programska izrada prototipa autonomnog vozila zasnovanog na mobilnoj platformi i postupcima strojnog učenja**

izrađen pod vodstvom mentora Prof.dr.sc. Goran Martinović

i sumentora Dino Kurtagić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD.....</b>	<b>1</b>
<b>1.1. Zadatak završnog rada .....</b>	<b>1</b>
<b>2. PREGLED STANJA U PODRUČJU AUTONOMNIH VOZILA.....</b>	<b>3</b>
<b>2.1. Povijesni pregled .....</b>	<b>3</b>
<b>2.2. Trenutno stanje u području .....</b>	<b>3</b>
2.2.1. Prednosti autonomnih vozila .....	6
2.2.2. Nedostaci autonomnih vozila .....	7
2.2.3. Komponente autonomnih vozila.....	7
2.2.4. Postojeći modeli Google Waymo i Tesla.....	9
<b>2.3. Izazovi upravljanja autonomnim vozilom .....</b>	<b>13</b>
<b>2.4. Maketa autonomnog vozila .....</b>	<b>14</b>
<b>3. PRIJEDLOG MODELA AUTONOMNOG VOZILA.....</b>	<b>15</b>
<b>3.1. Zahtjevi na autonomno vozilo.....</b>	<b>15</b>
3.1.1. Funkcionalni i nefunkcionalni zahtjevi na upravljački dio .....	15
3.1.2. Upravljački zahtjevi sustava .....	15
<b>3.2. Građa modela autonomnog vozila.....</b>	<b>15</b>
<b>3.3. Postupci strojnog učenja za upravljanje vozilom.....</b>	<b>16</b>
3.3.1. Primjena postupaka strojnog učenja .....	16
3.3.2. Prijedlog sklopovske arhitekture modela autonomnog vozila .....	17
<b>4. SKLOPOVSKI I PROGRAMSKI DIO RJEŠENJA AUTONOMNOG VOZILA.....</b>	<b>19</b>
<b>4.1. Sklopovske komponente .....</b>	<b>19</b>
4.1.1. Ultrazvučni senzori HC-SR04 .....	19
4.1.2. Sklop Croduino Basic3 .....	19
4.1.3. Upravljački modul za motor s dvostrukim H-mostom L298N .....	20
4.1.4. Kamera CMOS OV7670.....	21
<b>4.2. Maketa prototipa autonomnog vozila i njegove inačice .....</b>	<b>22</b>
4.2.1. Maketa modela prototipa autonomnog vozila.....	22
4.2.2. Prva inačica modela prototipa autonomnog vozila .....	22
4.2.3. Druga inačica modela prototipa autonomnog vozila .....	23
4.2.4. Ideja treće inačice modela prototipa autonomnog vozila.....	25

<b>4.3. Programsko implementiranje postupka strojnog učenja u prototip autonomnog vozila..</b>	<b>25</b>
<b>4.4. Programski dio rješenja za inačice prototipa autonomnog vozila.....</b>	<b>26</b>
4.4.1. Korišteni programski jezici i razvojna okolina.....	26
4.4.2. Programski kod za prvu inačicu prototipa autonomnog vozila.....	27
4.4.3. Programski kod za drugu inačicu prototipa autonomnog vozila.....	29
4.4.4. Programski kod za treću inačicu prototipa autonomnog vozila .....	33
4.4.5. Komunikacija sustava autonomnog vozila .....	33
<b>5. OPIS NAČINA KORIŠTENJA I ISPITIVANJE PROTOTIPA AUTONOMNOG VOZILA .....</b>	<b>34</b>
<b>5.1. Opis korištenja prototipa autonomnog vozila .....</b>	<b>34</b>
<b>5.2. Uvjeti ispitivanja i slučajevi korištenja .....</b>	<b>34</b>
5.2.1. Ispitni slučaj 1.....	34
5.2.2. Ispitni slučaj 2.....	34
5.2.3. Ispitni slučaj 3.....	35
5.2.4. Ispitni slučaj 4.....	35
5.2.5. Ispitni slučaj 5.....	36
5.2.6. Ispitni slučaj 6.....	36
<b>5.3. Analiza primijenjenog postupka strojnog učenja .....</b>	<b>37</b>
<b>5.4. Rezultati ispitivanja prototipa autonomnog vozila .....</b>	<b>37</b>
<b>ZAKLJUČAK.....</b>	<b>40</b>
<b>LITERATURA.....</b>	<b>41</b>
<b>SAŽETAK.....</b>	<b>45</b>
<b>ABSTRACT .....</b>	<b>46</b>
<b>ŽIVOTOPIS.....</b>	<b>47</b>
<b>PRILOZI (na DVD-u).....</b>	<b>48</b>

# 1. UVOD

Vožnja je izazov određen čimbenicima poput parametara vozila, ponašanja vozača i pješaka, fizičkih uvjeta, vremena i stanja površine ceste koje vozači moraju obraditi kako bi pravodobno donosili odluke i reagirali u stvarnom vremenu [1]. Proizvođači automobila razvijaju rješenja za umanjivanje izazova vožnje koja doprinose sigurnijoj i ugodnijoj pokretljivosti, a koja pak zahtijevaju vozila s funkcijom samostalne vožnje [1].

Cilj ovoga rada jest predložiti izradu sklopovskog i programskog dijela prototipa autonomnog vozila zasnovanog na mobilnoj platformi te primijeniti postupak strojnog učenja koji može podržati autonomno kretanje vozila. Glavni problem kojim se ovaj rad bavi je izbjegavanje prepreka na putu modela autonomnog vozila. Na taj problem utječu mnogi čimbenici, a između ostalog blizina prepreke, brzina očitavanja vrijednosti udaljenosti, vrsta prepreke, prostor u kojemu se model nalazi i drugi. Za rješavanje tog problema u prototip autonomnog vozila implementiran je pojednostavljeni postupak nadziranog strojnog učenja, odnosno klasifikacije sa svrhom predviđanja, a zasnovan je na stablu odluke. Rezultatima ispitnih slučajeva prikazuje se razina uspješnosti izbjegavanja prepreka, odnosno učinkovitost primijenjenog algoritma strojnog učenja.

Drugo poglavlje bavi se trenutnim stanjem u području autonomnih vozila te prikazuje prednosti, izazove i postojeće modelima. U trećem poglavlju definirani su zahtjevi na autonomno vozilo, njegova građa te su predloženi postupak strojnog učenja i sklopovska arhitektura modela. Četvrto poglavlje opisuje korištene sklopovske komponente, opisuje maketu modela kao i njegove inačice izvedbe. Također prikazano je programsko rješenje za svaku od predloženih inačica prototipa autonomnog vozila. U petom poglavlju opisan je način korištenja prototipa, provedeno ispitivanje za više slučajeva korištenja te analizirani dobiveni rezultati i model strojnog učenja.

## 1.1. Zadatak završnog rada

U teorijskom dijelu završnog rada potrebno je opisati i analizirati sklopovske i programske zahtjeve kojima se može omogućiti autonomna vožnja prototipa vozila. Nadalje, treba predložiti sklopovsko rješenje takvog sustava zasnovano na jednopločnom računalu i/ili mikroupravljaču (Raspberry Pi, Arduino), te odgovarajućim kamerama i sensorima. Također, treba osmisliti programsko rješenje koje omogućuje prikupljanje podataka s kamera i senzora, njihovu obradbu zasnovanu na odgovarajućim postupcima strojnog učenja s ciljem prepoznavanja objekata i prepreka, te upravljanja kretanjem vozila. Sklopovski dio rješenja treba ostvariti na nekoj od raspoloživih mobilnih platformi, a izgrađeni model strojnog učenja i upravljački program treba

implementirati u programsko rješenje u odgovarajućem programskom jeziku, razvojnoj okolini i uz korištenje odgovarajućih biblioteka. Prototip autonomnog vozila potrebno je ispitati s gledišta postupaka strojnog učenja i za različite uvjete, okolinu prepreka i scenarije vožnje.



## **2. PREGLED STANJA U PODRUČJU AUTONOMNIH VOZILA**

U ovom poglavlju opisuje se povijest i trenutno stanje u području autonomnih vozila, njihove prednosti i nedostatci, građa i izazovi kao i dva modela autonomnih vozila.

### **2.1. Povijesni pregled**

Prva konkretna ideja autonomnih vozila pojavljuje se dvadesetih godina prošloga stoljeća sa željom promjene uskih, loših cesta i omogućavanju brže kretanje i smanjivanju nesreća. Autonomnost, doduše, prvo se pojavljuje kod drugih vozila koja nemaju izloženost ogromnom prometu i izloženost velikom broju ljudi [2]. Autonomni automobili ostaju nerealizirana ideja do osamdesetih godina prošloga stoljeća kada znanstvenici razvijaju automatizirane sustave autocesta [3] [4]. Tim činom otvaraju mogućnost poluautonomnim i autonomnim vozilima za spajanje s infrastrukturom cesta.

Pionirski piloti autonomnih vozila napravljeni su u Njemačkoj i SAD-u od osamdesetih godina prošloga stoljeća do 2000. godine. 1980. godine Ernst Dickmanns i suradnici na Sveučilištu Bunderswehr u Münchenu dizajniraju Mercedes-Benz kombi koji doseže brzinu od 63 km/h na cestama bez prometa [5]. Nakon toga, EUREKA provodi projekt Prometheus od 1987. do 1995. na autonomnim vozilima. U istom desetljeću Defense Advanced Research Projects Agency (DARPA) financira Autonomous Land driven Vehicle (ALV) projekt koji ostvaruje prikladne rezultate. ALV projekt ostvaruje prvu demonstraciju praćenja ceste koristeći LIDAR, računalni vid i autonomnu robotsku kontrolu za usmjeravanje robotskog vozila brzinom do 31 km/h [6]. U lipnju 1993. profesor Han Min-Hong u Južnoj Koreji radi na autonomnom automobilu, koji uspjeva samostalno voziti čak 17 kilometara. Dvije godine kasnije, testiran je drugi automobil na autocesti Gyeongbu od Seoula do Busana 1995. godine [7].

S početkom 21. stoljeća publicitet autonomnim vozilima daje Google sa svojim autonomnim automobilom te privlači stručnjake iz svih područja. Današnja vodeća imena automobilske industrije su u utrci za monopolom i ostvarenju prvobitne ideje o potpuno autonomnom vozilu.

### **2.2. Trenutno stanje u području**

Autonomni automobili su ona vozila kojima upravljaju digitalne tehnologije bez ikakvog ljudskog djelovanja [8]. Sposobni su voziti se i kretati cestama prateći utjecaje na okoliš i okolinu te sa sposobnošću autonomnog odlučivanja i implementiranim strojnim učenjem donositi pravodobne odluke. Njihov izgled dizajniran je da zauzima manje prostora na cesti kako bi se izbjegle

prometne gužve i smanjila vjerojatnost nesreća. Unatoč razvoju i inovacijama, današnja vozila smatramo polupametnima. Mobilna ugrađena računala nude usluge bogate informacijama i zabavom sa senzorskim tehnologijama, primjerice kamere, GPS, detekcija i raspoređivanje svjetala (LIDAR, radar i druge tehnologije) kako bi bili svjesni svog okruženja upravljajući brzinom vozila, donošenjem odluka na temelju podataka o okolišu i kretanjem na sigurniji način. Osjetivši da bi takav fenomen mogao otvoriti novo tržište, velike tehnološke tvrtke Google, Apple i druge, prihvatile su se izazova izrade autonomnog vozila, s pratećim automobilskim tvrtkama. Razvoj autonomnih automobila s naprednim sustavima za izbjegavanje sudara i upravljanje prometom istraživanje je vrijedno 60.9 milijardi eura [9], a glavni proizvođači su Google i Tesla [10] [11] [12]. Projekti pametnih vozila svake se godine predstavljaju na sajmovima automobila CES, u salonu automobila u Ženevi, salonu automobila u Londonu i drugima.

Prema podacima Svjetske zdravstvene organizacije (engl. *World Health Organization, WHO*) u svijetu na cestama umre približno 1.35 milijuna ljudi svake godine, čiji smrtni slučajevi globalno koštaju 518 milijardi dolara. Prema istoj organizaciji, više od pola mortaliteta je među ranjivim sudionicima u prometu, odnosno pješacima, biciklistima i motociklistima [13]. Više je razloga zbog kojih dolazi do sudara, loša infrastruktura cesta i loš menadžment, mehanički problemi automobila, loši vremenski uvjeti i ponajviše ljudska pogreška (u više od 90% slučajeva) [14]. Tehnologija je bila glavni pokretač promjene gdje su inovacije dovele do značajnih transformacija [1], a trenutno automobilska industrija prolazi kroz dvije transformacije. Prva je postupni prelazak na obnovljive izvore energije ili pronalazak pogonskih sklopova koji će adekvatno zamijeniti dominaciju motora s unutarnjim izgaranjem. Druga transformacija je autonomna vožnja te danas gotovo svaki veći proizvođač automobila ima svoj program izrade autonomnog vozila [15]. S obzirom na početke i prvobitnu ideju autonomnih vozila dvadesetih godina prošloga stoljeća [2], iako je napredovanje ogromno, u 2017. dozvoljeni automatizirani automobili na javnim cestama nisu u potpunosti autonomni, svaki od njih treba ljudskog vozača koji će primijeti kada je potrebno vratiti kontrolu nad vozilom [8]. Autonomni automobili smatraju se rješenjem od kojeg se očekuje da smanji broj prometnih nesreća, troškove zagušenja i razinu onečišćenja, istovremeno povećavajući produktivnu upotrebu vremena provedenog u vozilu.

Danas su mnogi konvencionalni automobili već opremljeni s više sustava za pripomoć vozaču koji pomažu u vožnji i manevriranju, npr. tempomat (engl. *cruise control*), pomoć u traci, nadzor umora, pomoć pri parkiranju, prepoznavanje prometne signalizacije, kočenje u nuždi, upravljanje pri izbjegavanju, noćni vid, monitor mrtve točke, kamere koje pokrivaju 360°, pomoć u gužvi u prometu, i slično [15]. Integriranje različitih sustava za pripomoć vozaču dovodi do određenog

stupnja autonomije vožnje, što zahtjeva definiranje razina automatizacije. Važno je napomenuti da razina automatizacije može varirati od nule do potpune automatizacije. National Highway Traffic Safety Administration klasificira automatizaciju vozila od nulte do šeste razine [8] [16]:

- Bez automatizacije (razina 0) – Vozač u svakom trenutku ima potpuno i jedino zapovijedanje i kontrolu nad vozilom glede upravljanja, kočenja, gasa i motiva.
- Asistencija vozača (razina 1) – Neke određene funkcije upravljanja, poput elektroničkih kočnica, su automatizirane. Računalo nikada ne kontrolira upravljanje, ubrzanje ili naglo kočenje, jedino u nekim modelima vožnje [17]. Najbolji primjer za prvu razinu je prilagodljivi tempomat i pomoć pri parkiranju [8].
- Parcijalna automatizacija (razina 2) – Automatske su barem dvije upravljačke funkcije poput prilagodljivog tempomata u kombinaciji s centriranjem trake. Vozač i dalje mora obratiti pozornost na promet i biti u mogućnosti da preuzme kontrolu nad vozilom kada situacija zahtijeva, ali može maknuti ruke s volana [17].
- Kondicionalna automatizacija (razina 3) – Pod određenim prometnim uvjetima ili uvjetima okoline, vozač ustupa potpunu kontrolu nad svim ključnim sigurnosnim funkcijama i uvelike se oslanja na vozilo da pazi na bilo kakve promjene u uvjetima koji zahtijevaju prijelaz na kontrolu vozača. Od vozača će se zahtijevati da nastavi s nadzorom nad vozilom, ali s dovoljnim vremenom prijelaza [17]. Na ovoj razini automobil je sposoban odlučiti kada treba promijeniti traku i kako reagirati pri dinamičkim događajima na cesti, s vozačem kao sigurnosnim sustavom [8].
- Visoka automatizacija (razina 4) – Slična prethodnoj razini ali puno sigurnija. Vozilo se kreće propisno pod stabilnim okolnostima i ne treba ljudsku intervenciju. U slučaju upitne situacije zatražiti će pomoć korisnika, a ukoliko nema povratnu informaciju neće ugroziti putnike unutar vozila [8].
- Potpuna automatizacija (razina 5) – Vozilo je inteligentno dizajnirano za nadgledanje stanja na cesti i samostalno djelovanje, izvodeći sve sigurnosno-ključne funkcije vožnje tijekom cijelog putovanja (razina u potpunosti bez nadzora vozača) [8] [17].

Iako se s tehnološkog gledišta ne mogu definirati stroge granice razina autonomnosti, u pravilu se prva i druga razina mogu riješiti tradicionalnim algoritmima za obradu senzora s trenutno dostupnim sklopovskim procesorima pa se prema tome, one već nalaze u današnjim prodajnim

vozilima. Treća i četvrta razina se odnose na situacije u kojima se ruke vozača ne nalaze za volanom te je za te razine neophodna upotreba naprednog strojnog učenja [18].

### **2.2.1. Prednosti autonomnih vozila**

Iako je prijevoz sredstvo za poticanje prosperiteta društava, neizbježno je povezano s negativnim vanjskim eksternalizama poput zagađenja, nesreća i ljudskih žrtava. Postoji veliki broj studija koje procjenjuju ove troškove u vozilima kojima upravlja čovjek [19] [20]. Ovi se troškovi razlikuju od nastalih izravnih troškova poput troškova benzina, održavanja vozila, registracije vozila i licenciranja ili karata za javni prijevoz. Trošak eksternosti skriveni je trošak nametnut društvu u cjelini; uključuje troškove kao što su prometne zagušenja, nesreće i propadanje okoliša, kao i sigurnost. Općenito se smatra da tehnologija autonomnih vozila potencijalno može značajno umanjiti (ako ne i eliminirati) mnoge od ovih postojećih negativnih eksternalizama. Prema jednoj procjeni, ovi vanjski troškovi mogu biti visoki kao cijena goriva koja je nametnuta društvu kao cjelovita, na taj način uključujući pojedince s niskim prihodima koji se isključivo oslanjaju na javni prijevoz [21]. Autonomna vozila također mogu stvoriti dodatne prednosti poput povećanja pristupačnosti i mobilnosti pa čak i poboljšanja korištenja zemljišta.

Autonomna vozila ukazuju na vrednovanje vremena korisnika. S obzirom na to da pri potpunoj automatizaciji vozač u autu nema ulogu, ne dolazi do mentalnog niti fizičkog umora koje je rezultat vožnje, kao ni stresa. Sukladno tome, jedan od pozitivnih rezultata jest ušteda vremena i energije vozača, koje se mogu utrošiti na druge aktivnosti, bile one vezane uz posao ili zabavu, odnosno dolazi do poticaja produktivnosti. Još jedna od prednosti autonomnih vozila je i jednakost pred sustavom. Pošto vozač više ne igra ulogu u vožnji automobila, vozačke dozvole neće biti neophodne, što će mnogima omogućiti uštedu što se tiče prometa, a u isto vrijeme nitko neće biti diskriminiran što ne posjeduje vlastitu vozačku dozvolu. Isto će omogućiti i usmjeravanje ljudi prema novim radnim mjestima koja više neće ovisiti o osobnoj mobilnosti. Povećanjem kapaciteta cesta i učinkovitosti prometa može doći do smanjenja cestarine.

Pod uvjetom da je cijeli promet potpuno automatiziran, smanjiti će se i vjerojatnost prometnih nesreća, jer se smanjuje i vjerojatnost umora, naglih pokreta, zdravstvenih tegoba, utjecaja alkohola i droga, lančanih reakcija. Ukoliko s vremenom autonomna vozila prijeđu u potpunosti na električnu struju kao glavno gorivo umjesto benzina, doći će i do očuvanja okoliša, ili će se barem goriva koristiti pametnije i smanjiti emisiju [22]. Neka operativna ograničenja električnih automobila, uključujući kapacitet putovanja udaljenosti ograničen veličinom i trajnošću baterija te

proces pronalaska punionica, mogao bi se riješiti prirodnom i organskom ujedinjenošću električnih i autonomnih automobila [23] [24].

### **2.2.2. Nedostaci autonomnih vozila**

Jedan od glavnih nedostataka jest gubitak radnih mjesta osoba u prometnom sektoru poput vozača kamiona i autobusa, vozača taksija i sličnih. Dodatna edukacija i nadogradnja vještina neće riješiti probleme radnika, jer proširenje autonomnih automobila pogoršava opće društveno-ekonomske odnose, a rješenje za gubitak radnih mjesta nalazi se u restrukturiranju gospodarstva [25], što će utjecati na cijelu svjetsku gospodarsku sliku. Smanjenjem broja vozača i s mogućnošću korištenja autonomnog taksija također će doći do smanjenja i brojeva nesreća što će pogoditi tvrtke bazirane na popravku i održavanju automobila.

Jedno od najvećih problema je pitanje sigurnosti autonomnih automobila od mogućih kibernetičkih napada i zlouporabe podataka. Kao i svi elektronički uređaji, autonomna vozila su podložna napadima, što je potvrđeno 2015. godine kada dvojica hakera u eksperimentu preuzimaju kontrolu nad vozilom kroz sustav za zabavu i onemogućuju normalnu vožnju [26]. Godinu kasnije, isto se događa Teslinom modelu S [27]. S obzirom na mogućnost hakiranja vozila koja bi trebala biti u potpunosti sigurna, daljina istraživanja i nadogradnje su potrebne.

Dakako, s autonomnim automobilima postoji pravna, legalna dilema i rastuća etička pitanja. U slučaju nezgode ili sudara, postavlja se pitanje odgovornosti i pravilnog dodjeljivanja kazne, u situacijama ugrožavanja jednog života za spas mnogih važna je etika same odluke i hoće li isto vrijediti u svim situacijama, odnosno kada se treba ugroziti život vozača umjesto mnogih. Trenutno konkretnog odgovora nema, a daljnji pregovori su potrebni.

Autonomna vozila koriste razne senzorske sustave i digitalne mape za snalaženje u prostoru. Proces odabira pravilnih odluka odvija se neprestano u sekundi, a navigacijski sustav treba biti pouzdan u svim vremenskim uvjetima i trenutna pozicija vozila precizna i pouzdana u stvarnom vremenu. Problem nastaje kod optičkih sustava na kojima su tehnologije bazirane, i problemima prepoznavanja okoliša kada su skrivene trake na cestama, loši vremenski uvjeti, zasljepljujuće svjetlo od Sunca, zatamnjena stakla, geste rukama, blokirani GPS signal, itd. [15].

### **2.2.3. Komponente autonomnih vozila**

Pri kreiranju autonomnog automobila, potrebni su adekvatni razvojni alati, algoritmi i sklopovska podrška za procesiranje podataka i njihovo testiranje. Algoritmi se bave integracijom logičkih blokova u neovisnu programsku podršku za samostalno upravljanje, brojanje izmjeničnih napona

za rukovanje senzorima, obradu podataka, percepciju, lokalizaciju, donošenje odluka, usavršavanjem strojnog učenja i kontrolu nad vozilom [18]. Razvojni alati programske podrške alati su izvan mreže za rukovanje podacima tj. mreže za testiranje algoritama baziranim na strojnom učenju te njihova provjera i održavanje, dok sklopovlje predstavlja automobilski procesor visokih performansi a male snage koji omogućava optimizaciju algoritama niske razine sa specijaliziranom arhitekturom za izračune (zaključivanje) [18].

Uloga algoritama, u rješavanju tri glavna zadatka samoupravljanja, percepcije, lokalizacije i planiranja, pronaći je pravu ravnotežu između tradicionalnih algoritama zasnovanih na strojnom učenju [18]. Općenito je prihvaćeno da će strojno učenje biti pokretačka snaga programske podrške, iako specifična uloga usavršavanja strojeva i mjesto implementacije zauzimaju širok spektar, ovisno o korištenom pristupu [28]. Jedan od njih je primjena učenja s kraja do kraja (*engl. end-to-end*) promatranjem crne kutije, gdje automobil uči kako se kretati, pregovarati i prilagođavati upravljačke signale promatrajući ljudske vozače. Crna kutija, u strojnom učenju, se odnosi na funkciju ili sustav gdje se znaju i promatraju podaci u smislu ulaza i izlaza, ali se nema saznanja o unutarnjem radu funkcije ili sustava. Iako su u simuliranim okruženjima, za koje računalne igre pružaju učinkovitu razvojnu platformu, postignuti obećavajući rezultati, transparentnost sustava dovodi do otežanog praćenja grešaka, a testiranje sustava za svaki slučaj gotovo je nemoguće [18]. Robusniji, ali znatno složeniji način je analiza ljudske vožnje i definiranje građevnih elemenata programske podrške za samostalnu vožnju. Ovi blokovi strojnog učenja imaju hijerarhijski odnos i imaju namjeru pomoći glavnom zadatku pružanjem informacija o okolišu, interakciji vozilo-okolina i optimalnoj putanji koju treba izvesti [18]. Percepcija jest u tom slučaju spoj senzora na niskoj i visokoj razini, otkrivanje i klasifikacija objekata te rekonstrukcija apstraktnog okruženja, lokalizacija obuhvaća globalnu lokalizaciju i usmjeravanje, mapiranje, odometriju i lokalno pozicioniranje, dok planiranje podrazumijeva tumačenje scenarija, praćenje i predviđanje, planiranje kretanja (manevriranje), planiranje lokalne putanje i upravljanje aktorom.

U lancu opskrbe automobilske industrije, proizvođači automobila, kao proizvođači originalne opreme, igraju značajnu ulogu u spajanju komponenti, ugrađujući pojedinačne i neovisne komponente u konačni sustav. Prilagođeni alati za razvoj izvan mreže pružaju platformu za rukovanje podacima: prikupljanje podataka, anotacija (označavanje), generiranje i poboljšanje, prethodnu i naknadnu obradu i kalibraciju senzora, podršku algoritmima: prilagodljivo okruženje za obuku algoritama strojnog učenja i okvire za optimizaciju zaključivanja i spoj senzora na visokoj razini, odnosno udruženje različitih senzorskih podataka u stvarnim instancama, testiranje

koje obuhvaća provjeru preciznosti, opoziva, lažnog odbacivanja algoritama, razvoj *benchmarkinga* i mjernih podataka i složena simulacija bez mreže [18].

Iako su izvedbe algoritama nisko ograničene njihovim minimalno potrebnim faktorima pouzdanosti, raspoloživa procesorska snaga predstavlja još jedan izazov i gornju granicu složenosti što u slučaju masovne uporabe algoritama zasnovanih na strojnom učenju uglavnom utječe na sposobnost i brzinu zaključivanja u najsuvremenijim žičnim mrežama (NN). U slučaju konvolucijskih žičnih mreža (CNN), jednog od najučinkovitijih pristupa podatkovnih veza, to ograničava broj i veličinu slojeva, stvarajući potrebu za optimiziranim mrežnim arhitekturama i oslanjajući se na današnje ogromne najsuvremenije mreže. Računarstvo opće namjene je na grafičkim procesorskim jedinicama (GPU) standardni način obuke i inficiranja žičnih mreža za obradu senzorskih podataka i donošenje odluka koristeći prednosti optimiziranih strojeva za zaključivanje za masovno paralelno računanje procesorskih jedinica. Međutim, grafički procesori, kao računalne jedinice opće namjene, prvenstveno su optimizirani za izračunavanje „piksel po piksel“ na grafičkim strojevima, što ih čini nedovoljno optimalnim za obradu konvolucijskih žičnih mreža. Ova činjenica pokreće novu eru razvoja sklopovlja, pri čemu je dizajn integriranih sklopova sve više fokusiran na sklopovsko ubrzanje zaključivanja žičnih mreža, povećanje gustoće izvedbi ovih procesorskih jedinica i, zbog predviđene velike uporabe u automobilskoj industriji, dopuštanje integriteta sigurnosti automobila [18].

#### **2.2.4. Postojeći modeli Google Waymo i Tesla**

Velike tehnološke kompanije i one koje prednjače u automobilskim industrijama u konstantnoj su utrci za naslovom prvog u potpunosti autonomnog automobila. U ovom potpoglavlju će se pokriti dva modela u razvoju i ukratko način na koji funkcioniraju.

Počevši kao Google tim za razvoj samovozećeg automobila 2009. godine, Waymo je tehnološka kompanija orijentirana na strojno učenje i programsku podršku s ciljem proizvodnje samovozećeg automobila iako ne planiraju proizvesti vlastito vozilo [29]. Prvobitne inačice autonomnog vozila tvrtke Google oslanjaju se na vrlo detaljne mape cesta i terena ključne za određivanje pozicije automobila, za razliku od strogog određivanja pozicije GPS sustavima koji mogu imati razlikovnu pogrešku od nekoliko metara [30]. Slika 2.1 [31] prikazuje vozilo tvrtke Google.



Slika 2.1. Vozilo tvrtke Google [31].

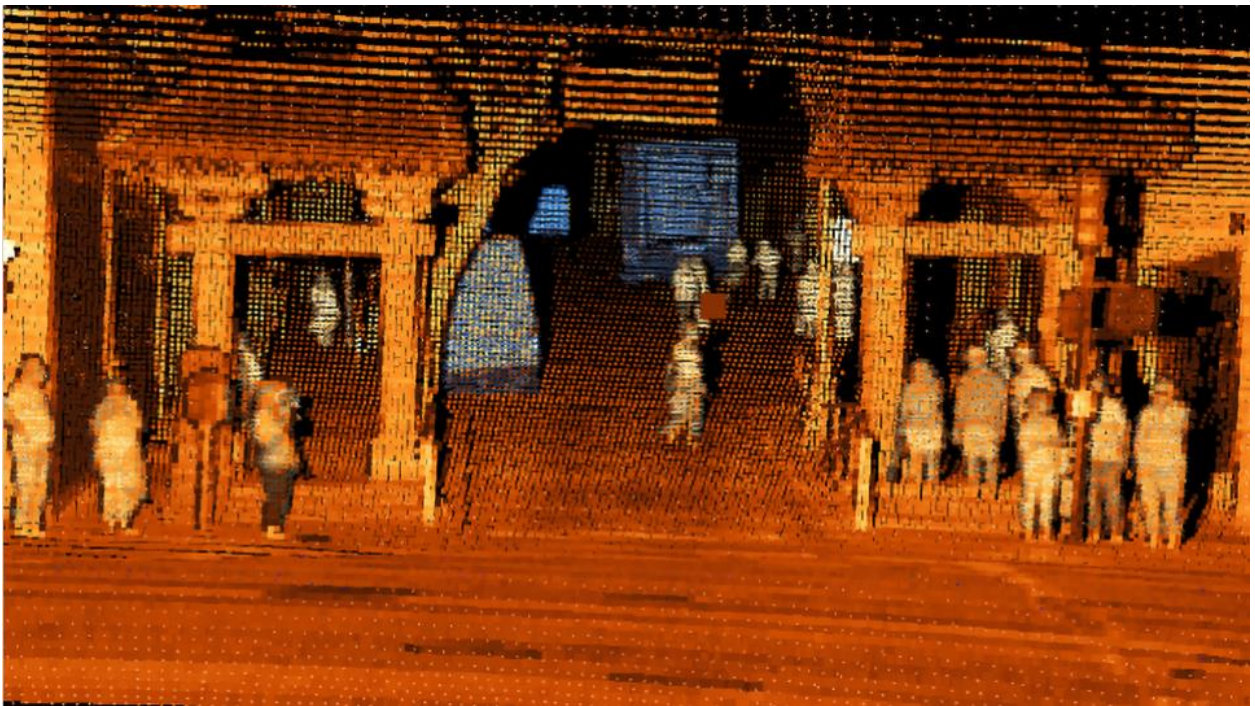
Vozilo tvrtke Google oslanja se na LIDAR (*eng. Light Detection and Ranging*) kako bi kreiralo 3D projekciju svoje okoline, a laser na krovu automobila predaje izmjerene vrijednosti i predaje ih procesoru koji stvara različite modele za upotrebu u prometu. Vozilo je također opremljeno s četiri radara postavljenih na prednje i stražnje odbojnice, kamerom blizu prednjeg retrovizora koja uočava semafore, GPS sustavom i koderom kotača za nadgledanje mobilnosti [30]. Slika 2.2 [31] prikazuje korištenje sustava LIDAR, dok slika 2.3 [31] prikazuje vizualizaciju podataka sustava LIDAR.



Slika 2.2. Korištenje LIDAR-a za kreiranje 3D slike okruženja [31].

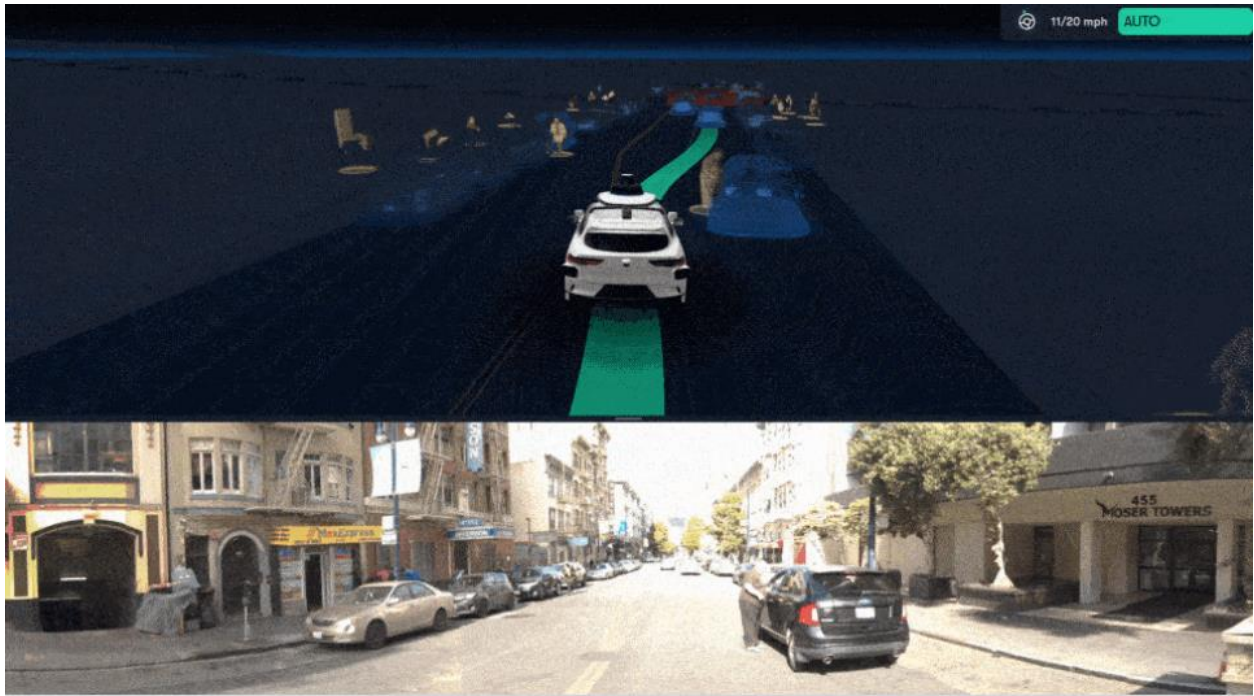


Tehnologija Waymo, dakako, je mnogo naprednija i prema podacima s njihove službene stranice, automobili opremljeni njome su prešli milijune kilometara na javnim cestama i milijarde u simulacijama [31]. Nakon desetljeća vožnje ulicama američkih gradova, Waymo uspijeva pravilno upravljati kroz gusti promet, uske ulice, raskrižja, kružne tokove i u bliskim susretima s neočekivanim pješacima, biciklistima i drugim korisnicima cesta. Razlog za veliki napredak su Waymovi napredni senzori koji grade kompleksan sustav za izbjegavanje nezgoda i pravilnu navigaciju prometom. Waymo koristi komplementarne senzore, radare, LIDAR i kamere za prikupljanje podataka, kao i njegov prethodnik. Razlika je što ima mogućnost gledanja 360° oko vozila tijekom dana i noći, kao i snalaženja u težim vremenskim uvjetima poput kiše i magle [32].



*Slika 2.3. Vizualizacija podataka sustava LIDAR [31].*

Kombinacija senzora u vozilu omogućava točno praćenje više objekata oko vozila prilikom vožnje. Zahvaljujući prikupljanju podataka s daleke udaljenosti i visoke gustoće samih slika, Waymo Driver može zaključiti s visokom točnošću zbog implementiranih algoritama udaljenosti manjih objekata i tuđim mobilnostima s veće udaljenosti, primjerice u slučaju zaustavljenog vozila i osobe koja polagano otvara vrata automobila da izađe iz njega. Za razliku od prethodnika s tradicionalnim automobilskim radarima, Waymov radar uočava statičke objekte i one u pokretu, što omogućava i sigurniju vožnju kroz maglu ili kišu [31]. Slika 2.4 [31] prikazuje očitavanja na sustavu za vrijeme vožnje.



*Slika 2.4. Očitavanja prilikom vožnje San Franciscom [31].*

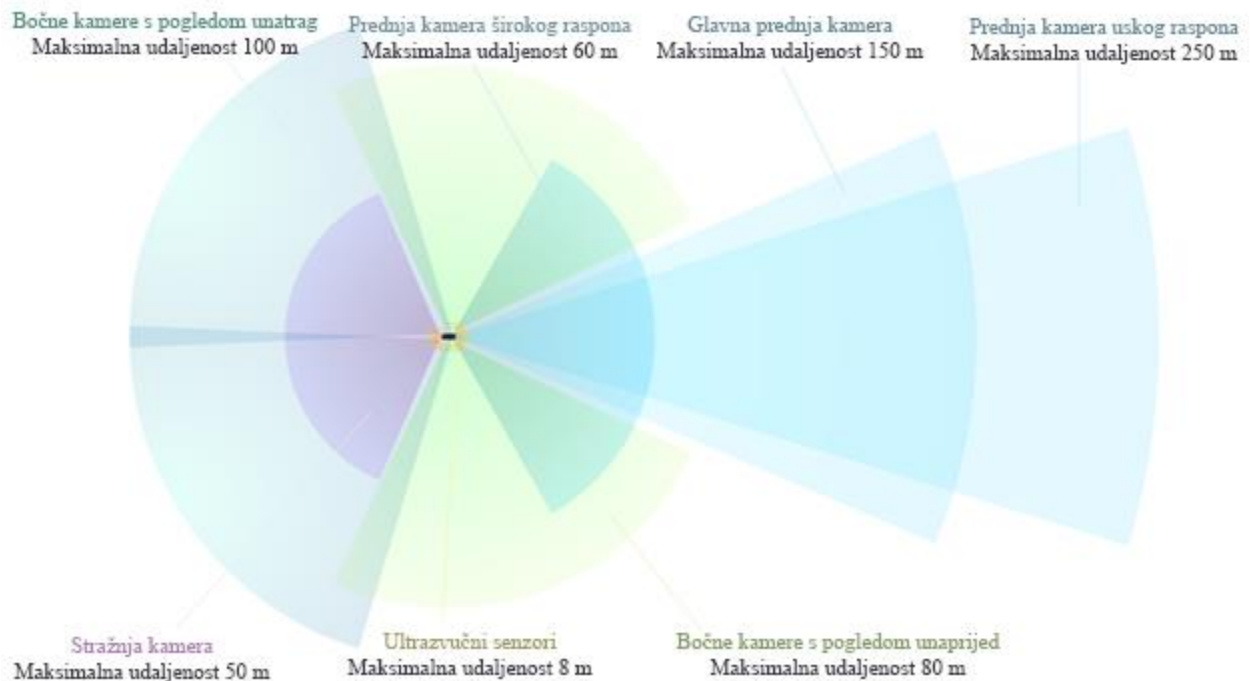
O autonomnim vozilima današnjice ne može se razgovarati bez spominjanja jedne od najvećih automobilskih tvrtki električnih vozila, Tesle. Osim poticanja obnovljive energije, Tesla godinama radi na mogućnostima opcijama autopilota i autonomne vožnje, iako su nadogradnje programske podrške dizajnirane da se njihova funkcionalnost postupno unaprijedi [33]. Slika 2.5 [34] prikazuje Teslin model S.



*Slika 2.5. Teslin model S [34].*

Teslina kombinacija senzora u vozilu omogućava pokrivanje od 360° i vidljivost oko vozila do 250 metara dometa. Tesla se oslanja na osam kamera i 12 ultrazvučnih senzora za pripomoć

vidokrugu koji omogućavaju primjećivanje tvrdih i mekih objekata na udaljenosti skoro dvaput veće od prijašnjih modela sustava [33]. Domet Teslinih senzora prikazan je na slici 2.6 [33].



Slika 2.6. Prikaz pokrića i domet senzora Teslinog autonomnog vozila [33].

Tri kamere postavljene iza vjetrobrana omogućuju široku vidljivost ispred vozila i fokusiranu detekciju objekata na većim udaljenostima. Glavna prednja kamera pokriva širok spektar usluga, prednja uska kamera bazira se na obradi objekata pri dalekim udaljenostima dok se prednje kamere koje gledaju postrance pod kutom od  $120^\circ$  fokusiraju na prometna svjetla i objekte koji naglo mogu presjeći putanju automobila. To je posebno korisno pri manevriranju urbanim područjima. Dvije kamere sa strane prate neočekivano priključivanje drugih sudionika u prometu u traku kojom se kreće autonomno vozilo, dvije retrovizorske kamere gledaju unatrag kako bi se pokrile mrtve točke, jedna retrovizorska kamera služi za ispomoc pri parkiranju i dodatnoj sigurnosti, dok ultrazvučni senzori učinkovito udvostručuju domet unaprijeđenom osjetljivošću s posebno kodiranim signalima [33].

### 2.3. Izazovi upravljanja autonomnim vozilom

Najveći izazovi upravljanja autonomnim vozilom su pronalazak prihvatljive i učinkovite logike strujnog učenja koje će osiguravati sigurnost sudionika u prometu kao i putnika u samome vozilu, algoritmi koji će u stvarnom vremenu na najbrži i najučinkovitiji način omogućiti kontrolu nad prikupljenim podacima i najidealniju odluku kretanja vozila, sklopovske komponente koje će biti

sukladne potrebama dijela programske podrške a davati najveću moguću kvalitetu obrade i prikupljanja podataka, sposobnost strojnog učenja da donosi moralne, za društvo najpogodnije, odluke u slučajevima ugrožavanja ljudskih života, mogućnost jednakog upravljanja autonomnim vozilima pri svim mogućim vremenskim uvjetima, i sigurnost autonomnih vozila od kibernetičkih napada. Naravno, pored izazova koji se odnose direktno na samo vozilo pri vožnji, postoje i izazovi pravne regulacije autonomnih vozila u prometu, adaptacija autonomnih vozila u društvo, unaprjeđivanje tehnologije i kamera kako bi se omogućila sigurna vožnja u svim vremenskim uvjetima, utjecaj na socioekonomskoj razini, itd.

## **2.4. Maketa autonomnog vozila**

Sukladno navedenim izazovima u prethodnom potpoglavlju, funkcionalna arhitektura autonomnog vozila zahtjeva obuhvaćanje podataka, procesiranje dobivenih podataka i pokretanje samog vozila. Predložen sustav kombinacija je više sklopovskih komponenti kamera, sustava LIDAR, senzora, GPS sustava i sustava RADAR [35] [36]. Visokorazvijeno snalaženje u prostoru i mogućnost vidljivosti 360° oko vozila povećava vjerojatnost uspješnosti pri izradi autonomnog vozila kao i pravilno uočavanje objekata oko njega. Zahtjevi pored snalaženja u prometu su i mogućnost promjene prometnog traka, parkiranje, zaobilaznje nezgoda, uočavanje objekata ili živih organizama na kratkim i dalekim udaljenostima, itd. Programska arhitektura sofisticiranijeg modela podrazumijeva nadzirano i nenadzirano učenje, pristupima klasifikacije, regresije, grupiranja, a neki od regresijskih algoritama korištenih za predikciju ishoda su, među ostalima, Bayesov algoritam, slučajna šuma, algoritam  $k$ -najbližih susjeda, grupiranje  $k$ -srednjih vrijednosti, višerazredne neuronske mreže [37].

### **3. PRIJEDLOG MODELA AUTONOMNOG VOZILA**

U ovom poglavlju definirani su funkcionalni i nefunkcionalni zahtjevi na upravljački dio prototipa autonomnog vozila, predložena je građa modela autonomnog vozila i postupak strojnog učenja zasnovan na algoritmu stabla odluke za upravljanje vozila.

#### **3.1. Zahtjevi na autonomno vozilo**

##### **3.1.1. Funkcionalni i nefunkcionalni zahtjevi na upravljački dio**

Mnogi su zahtjevi na programske komponente kao navigacijski model, mogućnost nadogradnje sustava, učinkovito strojno učenje, pravilni algoritmi, sposobnost umjetne inteligencije za donošenje pravilnih odluka u moralno upitnim situacijama, sigurnost od kibernetičkih napada. Glavni zahtjev na programske komponente koji se pokriva u ovome radu jest navigacijski model kojim se upravlja autonomnim vozilom, tj. sposobnost programa da obrađuje podatke i donosi odluke prema određenim podacima koje dobiva od komponenata koje čine vozilo. Taj zahtjev podrazumijeva i razvoj logike za mobilnost prototipa autonomnog vozila, kao i algoritme korištene unutar projekta, njihovu učinkovitost i složenost te slijednost radnji. Cilj je pronaći što jednostavniji i što uspješniji način samostalnog autonomnog upravljanja prototipa autonomnog vozila i omogućiti vozilu zaobilazanje prepreka i donošenje odluka u stvarnom vremenu. Isto tako, sekundarni zahtjev koji se obrađuje u ovome radu jest mogućnost nadogradnje sustava za upravljanje i povezivanje nadogradnji sa sustavom.

##### **3.1.2. Upravljački zahtjevi sustava**

Glavni zahtjev na upravljački dio jest povezanost sustava i ujedinjeni rad svih komponenata. Idući od zahtjeva je pravilno spajanje komponenata ovisno o njihovim zahtjevima. Primjerice, kamera CMOS OV7670 zahtijeva 3.3V dok sklop Croduino Basic3 zahtijeva 5V od baterije, gdje se sukladno tome treba pripaziti na voltažu predanu kameri CMOS. Jednako vrijedi za bilo koju komponentu prototipa autonomnog sustava i one koje bi se u budućnosti nadodale kao nadogradnja modela.

#### **3.2. Građa modela autonomnog vozila**

Sklopovska arhitektura osnovana je na mobilnoj platformi na koju se nadodaju potrebni dijelovi za izgradnju jednostavnog prototipa vozila. Mikroupravljač koji služi kao kontrolni centar modela je sklop Croduino Basic3 na koji se implementira algoritam stabla odluke i obrađuju podaci prikupljeni sa senzora, što znači da svi senzori moraju biti spojeni s mikroupravljačem.

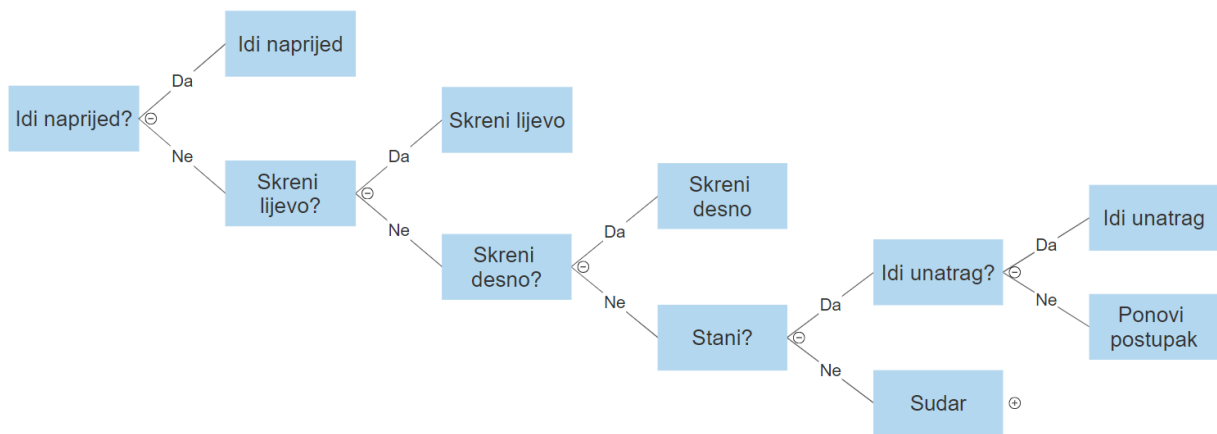
Mikroupravljač je također spojen sa motorima što se ostvaruje pomoću modula za upravljanje motora L298N s dvostrukim H-mostom. Tom povezanošću, mikroupravljač je u mogućnosti poslati naredbu motorima za kočenje, skretanje i nastavak kretanja naprijed. U slučaju kamere CMOS OV7670, ona mora prikazati sliku na spojeno računalo.

Programska arhitektura modela autonomnog vozila bazirana je na C++ programskom jeziku u Arduino IDE sučelju, preko kojeg se napisani kod šalje na mikroupravljač. Mikroupravljač iščitava predani algoritam i ukoliko su uvjeti zadovoljeni, izvršava odgovarajuću naredbu.

### **3.3. Postupci strojnog učenja za upravljanje vozilom**

#### **3.3.1. Primjena postupaka strojnog učenja**

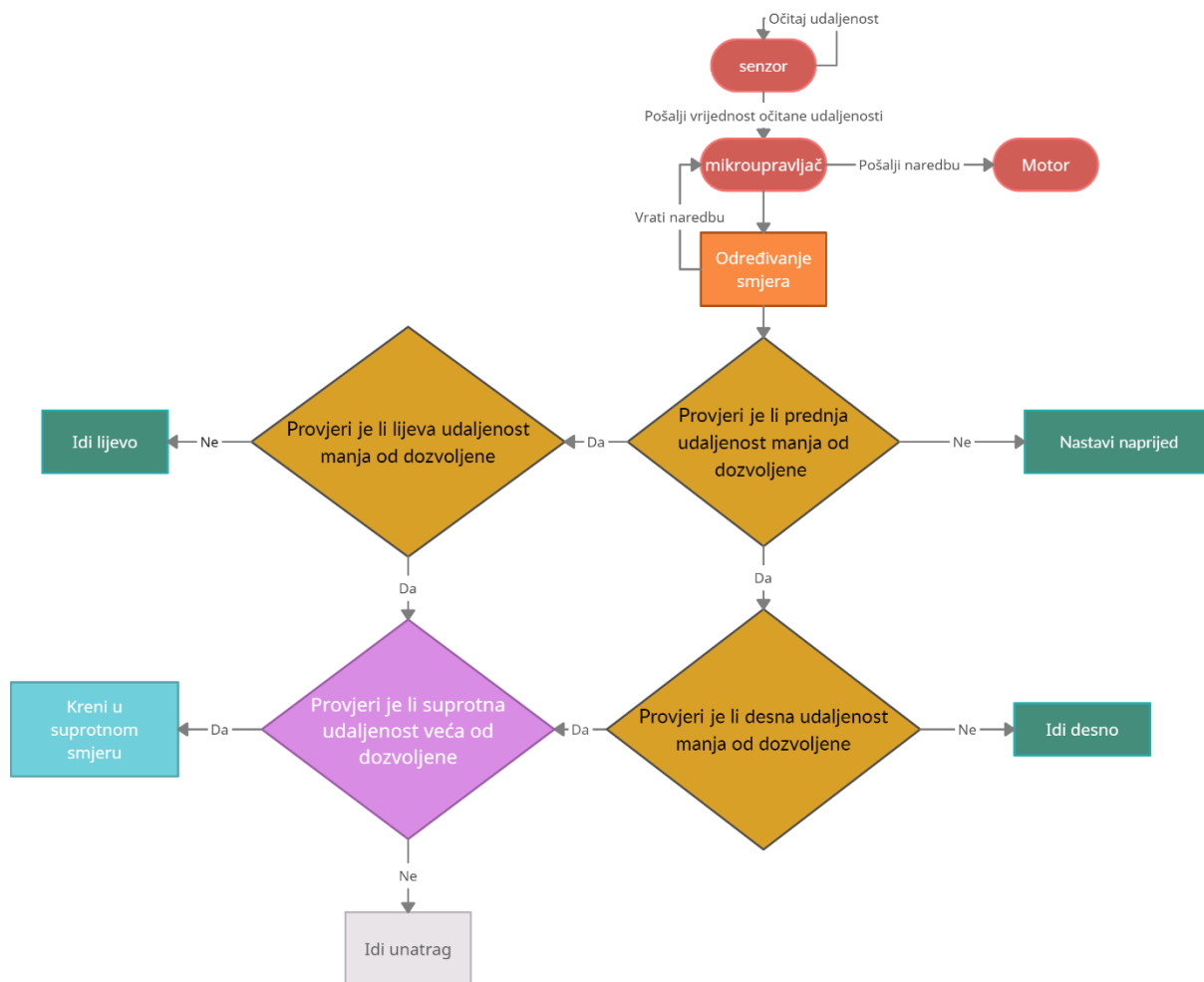
Strojno učenje primijenjeno u ovome završnom radu spada u nadzirno strojno učenje, tj. klasifikaciju u svrhu predviđanja. Primijenjeni algoritam strojnog učenja je algoritam stabla odluke, neparametarska diskriminativna metoda [38] koja se u ovome radu koristi kao klasifikator. Stablo odluke klasifikacija je primjera odozgo, odnosno od korijena prema listovima. Stablo se sastoji od čvorova, testnih atributa, i grane koja odgovara vrijednosti atributa [38]. Razlog odabira ovoga algoritma jest njegova jasnoća rezultata. Za algoritam primijenjen u radu, postoje dva ishoda svake situacije, prvi koje se izvodi ukoliko je testni atribut istinit za predanu vrijednost atributa, i drugi ishod ukoliko vrijedi suprotno. U ovome radu ti ishodi dolaze u obliku kretanja modela autonomnog vozila. Uzorci podataka prikupljeni sa senzora pohranjeni su u polja, predani u strukturu stabla gdje se uspoređuju vrijednosti udaljenosti i prema njima dolazi do pravilne odluke kretanja. Princip samoga stabla je jednostavan. Atribut koji je određen kao najbolji za kvalifikaciju ostalih atributa jest kretanje prema naprijed. Ukoliko je ono omogućeno, naredba za kretanje se poziva i vozilo se kreće sukladno odluci. U suprotnom se provjeravaju ostali atributi. Idući najvažniji postaje skretanje nalijevo te se kao i kod kretanja naprijed promatraju dva slučaja, kada je atribut istinit i kada je neistinit. U slučaju neistine prelazi se na idući čvor, skretanje prema desno i ukoliko je i ono onemogućeno u obzir dolazi zaustavljanje vozila. U slučaju neistine vozilo ostaje u pokretu i dolazi do sudara, u slučaju istine preispituje se posljednji testni atribut, kretanje unatrag. U slučaju neistine, vozilo je zaustavljeno od prijašnjeg čvora i ponavlja se postupak. U slučaju istinitosti, poziva se naredba za kretanje unatrag za izbjegavanje sudara. Iako jednostavan model algoritma stabla odluke, u svrhu ovoga rada odabran je kao polazni algoritam strojnog učenja. Slika 3.2 prikazuje opisani algoritam stabla odluke.



Slika 3.2. Algoritam primijenjenog stabla odluke.

### 3.3.2. Prijedlog sklopovske arhitekture modela autonomnog vozila

Ideja algoritma jest da se autonomno vozilo pri primanju signala o radu motora krene kretati prema naprijed. Ukoliko autonomno vozilo pri samom početku vožnje ima objekt ili prepreku pred sobom, mora provjeriti udaljenosti s lijeve i desne strane i skrenuti prema onoj gdje je udaljenost veća, tj. prema udaljenosti čija se iduća prepreka nalazi dalje od vozila ili uopće ne postoji. Model autonomnog vozila se treba kretati u prostoru s mogućnošću navigacije između objekata, njihovom zaobilazanju i po potrebi, kretanju unatrag ukoliko se nalazi u situaciji gdje skretanje lijevo ili desno, ili kretanje naprijed, rezultira sudarom. Očitavanja sa senzora su ključna za pravilnu navigaciju prostorom, dok kamera u sklopu rada prikazuje sliku okoline u kojoj se vozilo nalazi. Projektirana sklopovska i programska arhitektura predloženoga modela prikazana je na slici 3.3.



Slika 3.3. Projektirana sklopovska i programska arhitektura modela autonomnog vozila.



## 4. SKLOPOVSKI I PROGRAMSKI DIO RJEŠENJA AUTONOMNOG VOZILA

U ovom poglavlju objašnjena je građa prototipa autonomnog vozila, njegove komponente i razvoj prototipa kroz tri inačice. Također, objašnjeno je programsko rješenje za obradu podataka sa senzora koje podrazumijeva primjenu nadzirnog strojnog učenja, odnosno klasifikacije s ciljem predviđanja na temelju implementiranog algoritma stabla odluke.

### 4.1. Sklopovske komponente

#### 4.1.1. Ultrazvučni senzori HC-SR04

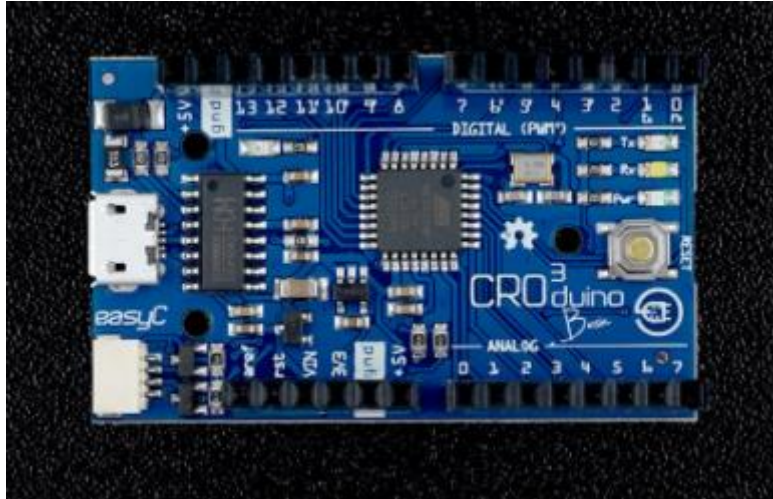
Ultrazvučni senzor HC-SR04 prikazan na slici 4.1 [39] senzor je za udaljenost s mogućnošću razlikovanja od 2 cm do 400 cm bez kontaktne mjerne funkcionalnosti s preciznošću do 3mm i mjernim kutom od 30°. Svaki ultrazvučni senzor sadrži ultrazvučni odašiljač, prijamnik i upravljački krug. Posjeduje četiri nožice za spajanje, VCC (napon), Trig (okidač), Echo (prijamnik) i GND (uzemljenje) [39].



Slika 4.1. Prikaz ultrazvučnog senzora HC-SR04 [39].

#### 4.1.2. Sklop Croduino Basic3

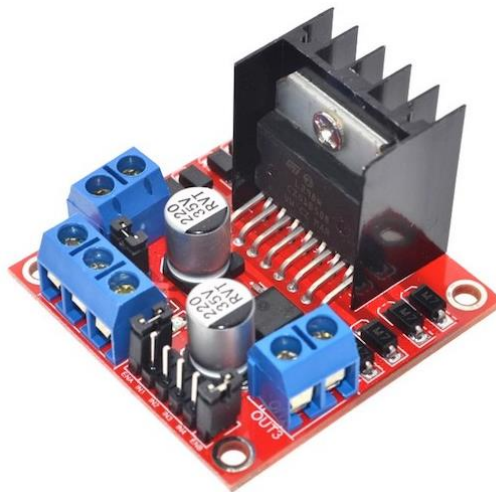
Sklop Croduino Basic3 hrvatska je Arduino kompatibilna pločica. Na sebi ima mikroupravljač Atmelov Atmega328 koji omogućava dvadeset i dvije I/O nožice, dok se za komunikaciju između pločice i računala koristi USB CH340 prema mostu UART. Na pločici se nalazi konektor easyC, koji omogućava jednostavno povezivanje raznih uređaja s pločicom. Na trinaestoj nožici nalazi se LED, a Croduino Basic3 posjeduje i regulator za mogućnost vanjskog napajanja i zaštitni osigurač [40]. Na slici 4.2 prikazan je sklop Croduino Basic3 [40].



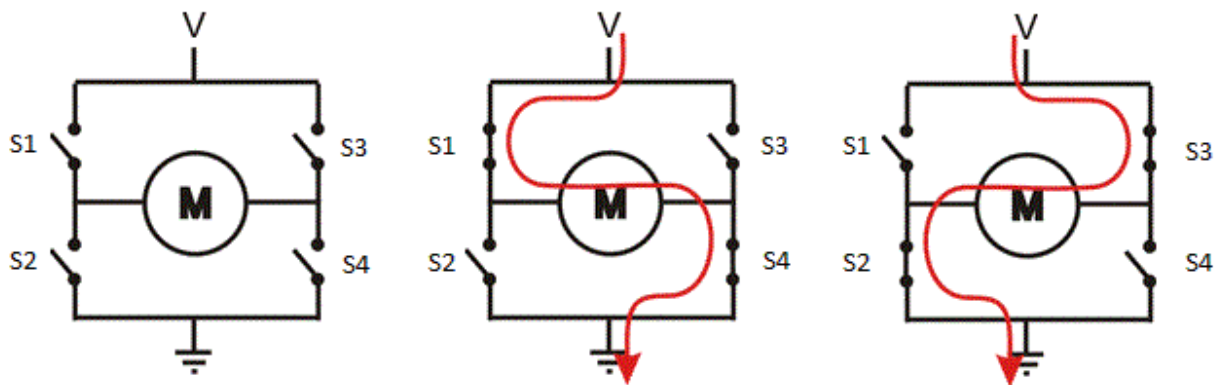
Slika 4.2. Prikaz sklopa Croduino Basic3 [40].

#### 4.1.3. Upravljački modul za motor s dvostrukim H-mostom L298N

Upravljački modul za motor L298N prikazan na slici 4.3 [41] koristi integrirani sklop H-most (*engl. H-bridge*). H-most strujni je krug od četiri prekidača. Način spajanja prekidača s motorom prikazan je na slici 4.4. Ako su prekidači S2 i S3 zatvoreni, a S1 i S4 otvoreni, struja će teći u određenom smjeru. Ukoliko su prekidači spojeni obrnuto, struja teče motorom u suprotnom smjeru. Zatvaranje sva četiri prekidača odjednom ili dva prekidača na jednoj strani H-mosta rezultira kratkim spojem [41]. Slika 4.4 [41] prikazuje opisano spajanje prekidača s motorima.



Slika 4.3. Prikaz upravljačkog modula za motor L298N [41].



Slika 4.4. Prikaz spajanja prekidača s motorima [41].

#### 4.1.4. Kamera CMOS OV7670

Kamera CMOS OV7670 niskonaponski je senzor slike koji pruža potpunu funkcionalnost kamere VGA s jednim integriranim sklopom i procesora slike u malom paketu otisaka. OV7670 pruža osam bitni prikaz u punom kadru ili u prozoru slike u širokom rasponu formata, kojima se upravlja putem sučelja sabirnice za upravljanje serijskom kamerom (SCCB). Kamera ima niz slika sposobnih za rad do 30 sličica u sekundi (fps) u VGA-i s potpunom korisničkom kontrolom kvalitete slike, oblikovanja i prijenosa izlaznih podataka. Sve potrebne funkcije za obradu slike, uključujući kontrolu ekspozicije, gamu, ravnotežu bijele, zasićenje boja, kontrola nijansi i drugo, također se mogu programirati putem sučelja SCCB. Kamera je rezolucije 640x480 VGA, a potrebno joj je samo 3.3V za rad. Izlaz s kamere podržava formate Raw RGB, RGB (GRB 4:2:2, RGB565/555/444), YUV (4:2:2) i YCbCr (4:2:2) [42]. Slika 4.5 prikazuje izgled kamere CMOS OV7670 [42].



Slika 4.5. Prikaz kamere CMOS OV7670 [42].

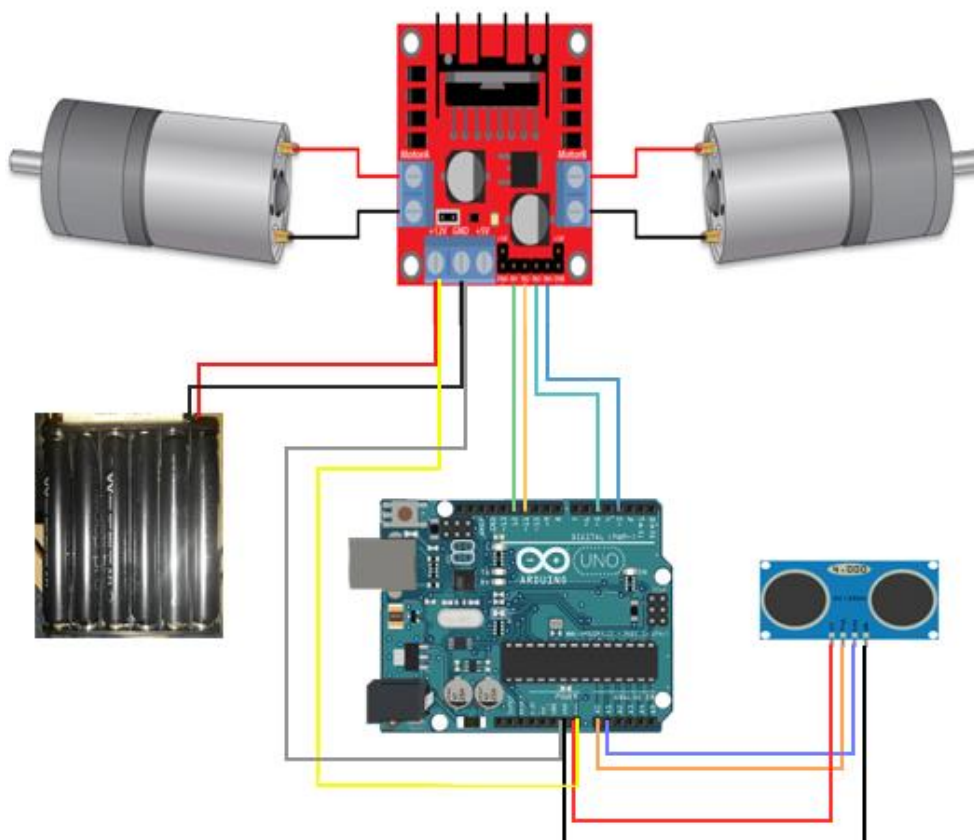
## **4.2. Maketa prototipa autonomnog vozila i njegove inačice**

### **4.2.1. Maketa modela prototipa autonomnog vozila**

Maketa modela razvijana je kroz dvije inačice s idejom za treću. Prva inačica koristi se jednim senzorom, druga s tri senzora i treća inačica ima ugrađenu kameru CMOS OV7670 na mobilnu platformu. Razlog više inačica je postupno uočavanje problema ograničenosti sklopovskog i programskog dijela modela te pokušaj njegove nadogradnje i razvoja algoritma strojnog učenja. Za izradu sklopovskog dijela prototipa modela autonomnog vozila korištena su tri ultrazvučna senzora HC-SR04 za snalaženje u prostoru i očitavanje udaljenosti vozila od prepreka, sklop Croduino Basic3 kao mikroupravljač, upravljački modul za motor L298N s dva H-mosta za upravljanje brzinom i smjerom vrtnje motora, baterija od 7.6 V i kamera CMOS OV7670 čija se slika prenosi na računalo. Prototip korišten za ispitivanje autonomnog vozila prikazan na slici 4.6 je bez instalirane kamere koja je ograničena dosegom povezivanja USB kabelom, što uvelike smanjuje mobilnost prototipa autonomnog vozila. Autor prepoznaje potrebu kamere CMOS OV7670 za unapređivanje modela i funkcionalnosti te s tom idejom radi na trećoj inačici koja treba pokazati uspješno povezivanje kamere CMOS OV7670 s računalom. Treća inačica nije mobilna zbog prekratkog USB kabela te je cilj spajanje autonomnog vozila s računalom bez dodatne obrade podataka, iako autor smatra kameru kao veliku nadogradnju za budući model prototipa autonomnog vozila i mogućnost implementiranja više algoritama strojnog učenja.

### **4.2.2. Prva inačica modela prototipa autonomnog vozila**

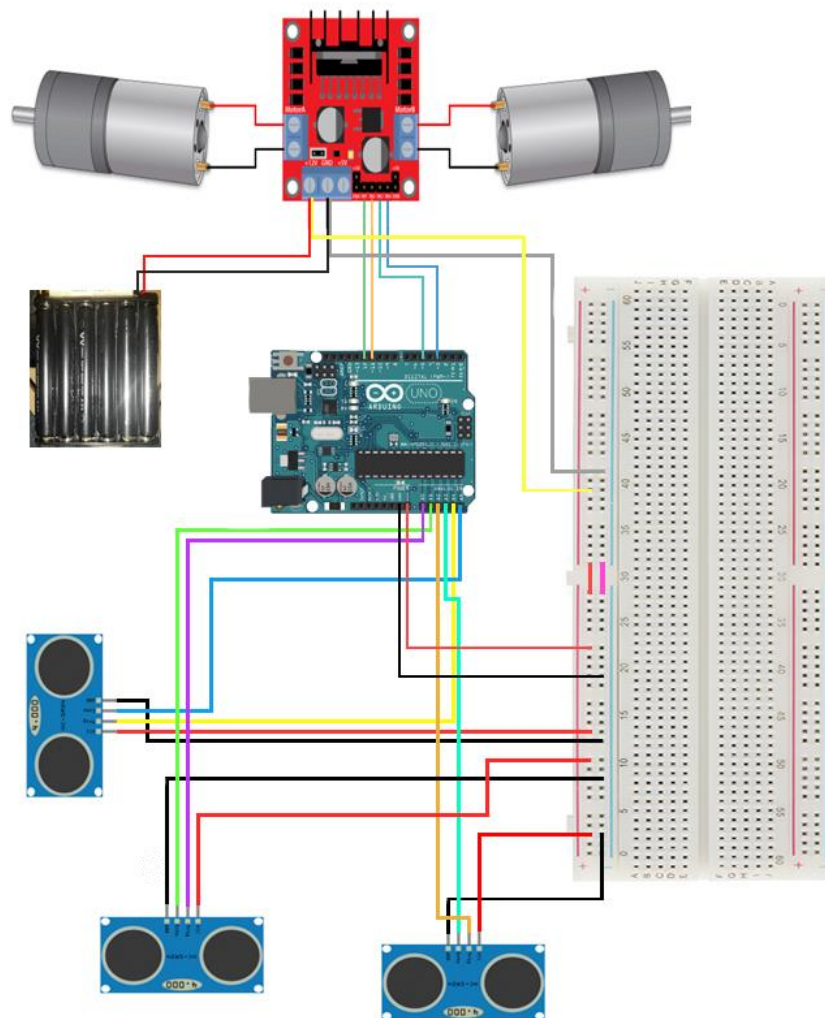
Prva inačica sklopovskog dijela prototipa modela autonomnog vozila sadrži jedan ultrazvučni senzor HC-SR04, sklop Croduino Basic3, upravljački modul za motor L298N za upravljanje brzinom i smjerom vrtnje motora i bateriju od 7.6 volti. Slika 4.5 prikazuje shematski prikaz sklopa prve inačice prototipa.



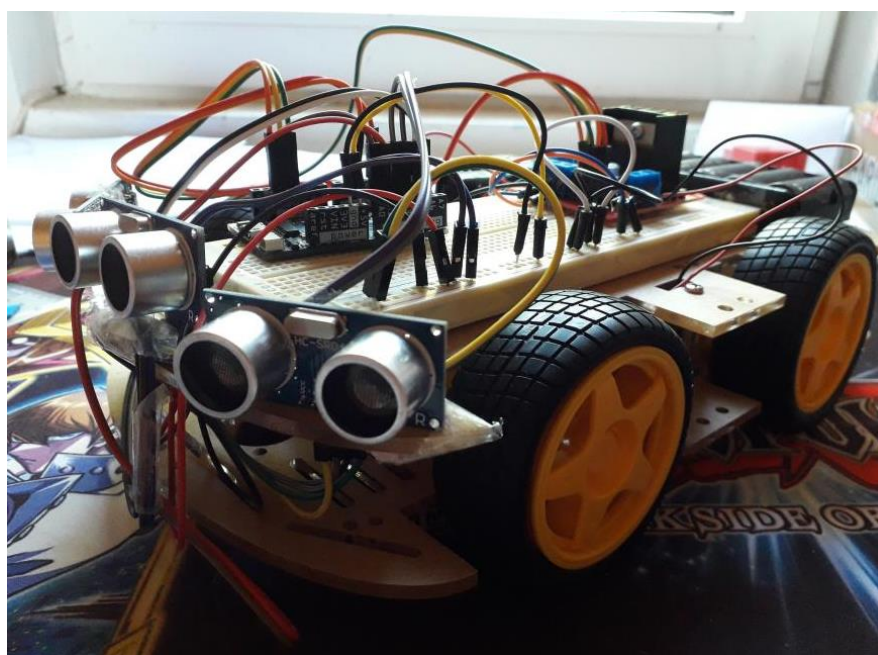
*Slika 4.5. Shema prve inačice sklopa prototipa modela autonomnog vozila.*

#### **4.2.3. Druga inačica modela prototipa autonomnog vozila**

Druga inačica sklopovskog dijela modela autonomnog vozila nadogradnja je prve gdje se broj senzora povećava s jedan na tri, u maketu se uključuje protoboard, dok ostale komponente, jedan ultrazvučni senzor HC-SR04, sklop Croduino Basic3, L298N H-most za upravljanje brzinom i smjerom vrtnje motora i baterija od 7.6 volti, ostaju jednake. Slika 4.6 prikazuje shemu druge inačice prototipa modela autonomnog vozila, dok slika 4.7 prikazuje laboratorijsku izvedbu sheme.



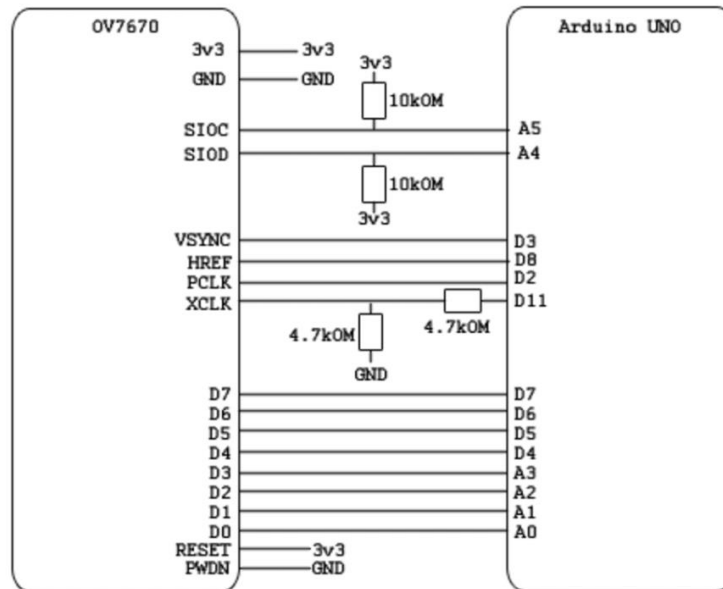
Slika. 4.6. Sklopovska shema druge inačice prototipa modela autonomnog vozila.



Slika. 4.7. Izvedba sklopovskog dijela prototipa modela autonomnog vozila.

#### 4.2.4. Ideja treće inačice modela prototipa autonomnog vozila

Treća inačica nastaje s idejom jednakih komponenata kao i druga inačica, s dodatkom kamere CMOS OV7670. Kamera CMOS OV7670 povezana je sa sklopom Croduino Basic 3 prema shemi na slici 4.8 [43].



Slika 4.8. Shema spajanja kamere CMOS OV7670 sa sklopom Croduino Basic 3 [43].

### 4.3. Programsko implementiranje postupka strojnog učenja u prototip autonomnog vozila

Strojno učenje odvija se i unaprjeđuje kroz drugu inačicu prototipa autonomnog vozila. Prva inačica prototipa autonomnog vozila na koju je implantiran jednostavan programski kod *autonomous\_car\_1.ino* koristi se bibliotekom `<HCSR04.h>` za dohvaćanje vrijednosti udaljenosti jednoga senzora HC-SR04 u centimetrima te prema dohvaćenoj vrijednosti ulazi u *loop()* u kojem se provjerava ukoliko je udaljenost manja od -1 (jer su zabilježena očitavanja u negativnim vrijednostima) ili 25 centimetara, te ukoliko je to istinito, model autonomnog vozila se zaustavlja naredbom *stop()*, radi stanku, kreće se unatrag pozivom naredbe *goBackward()* kako bi oslobodio prostor za pravilno skretanje, staje, i slučajno odabire hoće li se kretati lijevo te pozvati naredbu *goLeft()*, ili desno pozivom naredbe *goRight()*. Ukoliko ne postoji prepreka pred prototipom autonomnog vozila, poziva se metoda *goForward()* i autonomno vozilo kreće se prema naprijed.

Prva inačica jednostavna je inačica logike korištene za kretanje modela autonomnog vozila koja ima velik prostor za nadogradnju. S tom idejom i potrebom konkretnije obrade podataka s više od jednoga senzora te mogućnošću odabira smjera s razlogom, a ne slučajno kao u prvoj inačici,

nakon mnogih testiranja i unapređivanja koda, nastaje konačna inačica programa implementiranog na drugu inačicu prototipa autonomnog vozila, *autonomous\_car\_final.ino*.

Druga inačica prototipa autonomnog vozila na kojem je implementiran programski kod *autonomous\_car\_final.ino* prati logiku jednostavnog algoritma stabla odluke za kretanje u prostoru. Biblioteka *<HCSR04.h>* pokazuje se ograničenom s dohvaćanjem podataka više od jednog senzora te se prelazi na novu, prikladniju biblioteku *<NewPing.h>*. Pomoću nove biblioteke jednostavno je dohvaćanje podataka triju senzora HC-SR04 te prema njihovoj obradi dolazi do određene, pravilne akcije. Vrijednosti dohvaćene sa senzora pohranjuju se u niz *sensors[]* te ovisno o pročitanim vrijednostima, mikroupravljač šalje naredbu o idućoj akciji autonomnog vozila prema motorima koji je izvršavaju. Ideja logike jednostavnog stabla odluke jest da ukoliko se ispred prototipa autonomnog vozila nalazi objekt tj. prednji senzor zabilježi vrijednost manju ili jednaku 25 (što je minimalna dozvoljena udaljenost između objekta i modela autonomnog vozila), algoritam stabla odluke provjerava očitane udaljenosti lijevog i desnog senzora te donosi odluku. Ukoliko se s lijeve strane nalazi objekt, tj. udaljenost je manja ili jednaka 25, vozilo treba skrenuti nadesno naredbom *goRight()*, i obrnuto, ukoliko se s lijeve strane nalazi objekt, vozilo skreće nalijevo naredbom *goLeft()*. Ukoliko se niti s lijeve niti s desne strane ne nalazi prepreka, algoritam treba usporediti očitane vrijednosti udaljenosti s obaju senzora i dati prednost smjeru koji ima veću vrijednost udaljenosti. U slučaju da se prototip autonomnog vozila nalazi u situaciji gdje su očitavanja s triju senzora ispod dozvoljenih, izvršava se naredba *goBackward()* za kretanje unatrag dok se model ne nađe u situaciji da može skrenuti ili nastaviti ravno. U slučaju da model autonomnog vozila nema nikakvih prepreka ispred sebe, izvršava se naredba *goForward()* za kretanje unaprijed.

Treća inačica prototipa autonomnog vozila nadogradnja je druge te je za snalaženje u prostoru implementiran identičan kod druge inačice s dodatkom koda za dohvaćanje slike kamere. Kamera pri samom spajanju s računalom prikazuje sliku na grafičkom korisničkom sučelju *Serial Port Reader* i može spremirati sliku na računalo.

## **4.4. Programski dio rješenja za inačice prototipa autonomnog vozila**

### **4.4.1. Korišteni programski jezici i razvojna okolina**

U svrhu rada korišteno je sučelje Arduino IDE, čiji je glavni programski jezik C/C# i biblioteka *<HCSR04.h>* za jedan senzor HC-SR04 i biblioteka *<NewPing.h>* za korištenje triju senzora, dok je za kameru CMOS OV7670 korišteno pet biblioteka: *<stdint.h>*, *<avr/io.h>*, *<util/twi.h>*, *<util/delay.h>* i *<avr/pgmspace.h>*.



#### 4.4.2. Programski kod za prvu inačicu prototipa autonomnog vozila

Programski kod započinje uključivanjem biblioteke <HCSR04.h> potrebne za očitavanje udaljenosti jednoga senzora HC-SR04 te inicijalizacijom senzora u programski kod. Također, inicijalizirani su nožice lijevog i desnog motora, te varijabla *distance* realnog tipa u koju će se pohranjivati očitavanja udaljenosti sa senzora. Nožice brzine spojene su na istu nožicu za kretanje naprijed jer u suprotnom dolazi do problema s upravljačkim modulom za motor L298N. Slika 4.9 prikazuje opisani kod.

```
1  #include <HCSR04.h>
2  UltrasonicDistanceSensor ultrasonic(A0,A1);
3  float distance;
4  // left motor
5  int leftMotorSpeedPin = 3;
6  int leftMotorForwardPin = 3;
7  int leftMotorBackwardPin = 5;
8  // right motor
9  int rightMotorSpeedPin = 11;
10 int rightMotorForwardPin = 11;
11 int rightMotorBackwardPin = 12;
```

Slika 4.9. Inicijalizacija senzora i motora.

Slijede postavke nožica motora da se ponašaju kao izlazi naredbama *pinMode()* čija je sintaksa *pinMode(pin, mode)*. Postavlja se brzina serijalnog prijenosa podataka na 9600 bita po sekundi naredbom *Serial.begin(9600)*, dok naredba *digitalWrite()* sintakse *digitalWrite(pin, value)* postavlja vrijednosti predanih nožica motora na HIGH od vrijednosti 5 V ukoliko ih se želi uključiti, ili na LOW u vrijednosti od 0 V ukoliko trebaju ostati ugašeni. Navedeno je prikazano na slici 4.10.

```
13 void setup() {
14     pinMode(leftMotorSpeedPin, OUTPUT);
15     pinMode(leftMotorForwardPin, OUTPUT);
16     pinMode(leftMotorBackwardPin, OUTPUT);
17     pinMode(rightMotorSpeedPin, OUTPUT);
18     pinMode(rightMotorForwardPin, OUTPUT);
19     pinMode(rightMotorBackwardPin, OUTPUT);
20     Serial.begin(9600);
21     digitalWrite(leftMotorSpeedPin, HIGH);
22     digitalWrite(leftMotorForwardPin, HIGH);
23     digitalWrite(leftMotorBackwardPin, LOW);
24     digitalWrite(rightMotorForwardPin, HIGH);
25     digitalWrite(rightMotorBackwardPin, LOW);
26     digitalWrite(rightMotorSpeedPin, HIGH);
27 }
28 }
```

Slika 4.10. Postavke nožica motora.

Nakon toga slijedi petlja *loop()*, odnosno petlja u kojoj se nalaze naredbe i algoritam kretanja prototipa autonomnog vozila. Dohvaća se očitavanje vrijednosti udaljenosti senzora HC-SR04 i pohranjuje u varijablu *distance*. Naredba *Serial.println(distance)* omogućava praćenje vrijednosti udaljenosti na Arduino IDE Serial Monitoru. *If* izraz prati logiku opisanu u potpoglavlju ranije. Ukoliko je očitana udaljenost veća od -1 (zbog negativnih očitavanja vrijednosti), i udaljenost manja od 25 (dozvoljene udaljenosti između vozila i prepreke), poziva se naredba *stopCar()*, tj. stani, pričekaj 750 milisekundi pomoću naredbe *delay(750)*, vrati se unatrag naredbom *goBackward()*, idi unatrag 300 ms, stani, pričekaj 700 ms. Ulazi se u novi *if* izraz koji kaže, ako je slučajni odabir broja pozvan naredbom *random()* jednak 0, skreni lijevo naredbom *goLeft()*, u suprotnom izvrši skretanje nadesno naredbom *goRight()*, ponovno stani. Ako prvi slučaj nije istinit idi naprijed naredbom *goForward()*. Opisani kod prikazan je na slici 4.11.

```

29 void loop() {
30     distance = ultrasonic.measureDistanceCm();
31     Serial.println(distance);
32     if (distance > -1 && distance < 25) {
33         stopCar();
34         delay(750);
35         goBackward();
36         delay(300);
37         stopCar();
38         delay(700);
39
40         if(random(0, 2) == 0) {
41             goLeft();
42         } else {
43             goRight();
44         }
45
46         delay(500);
47         stopCar();
48         delay(700);
49     }
50     else {
51         goForward();
52     }
53 }

```

Slika 4.11. Prikaz petlje *loop()*.

Zadnji dio programskog koda obuhvaća naredbe *goForward()*, *goBackward()*, *stopCar()*, *goRight()* i *goLeft()*, tj. postavke nožica naredbom *digitalWrite()* na HIGH ili LOW ovisno o njihovoj ulozi. Primjerice, za kretanje naprijed, i lijevi i desni motori moraju biti uključeni, tj. nožice biti podešene na HIGH, dok nožice za kretanje unatrag moraju biti isključene, tj. biti postavljene na LOW, za unatrag vrijedi obrnuto. Sukladno toj logici, kada autonomno vozilo staje, sve nožice moraju biti isključene, kao što za kretanje nadesno lijeva prednja nožica je uključena a ostale isključene. Za skretanje nalijevo vrijedi suprotno. Naredbe za kretanje i postavke nožica motora prikazane su na slici 4.12.

```

54 void goForward() {
55     digitalWrite(leftMotorForwardPin, HIGH);
56     digitalWrite(leftMotorBackwardPin, LOW);
57     digitalWrite(rightMotorForwardPin, HIGH);
58     digitalWrite(rightMotorBackwardPin, LOW);
59 }
60 void goBackward() {
61     digitalWrite(leftMotorForwardPin, LOW);
62     digitalWrite(leftMotorBackwardPin, HIGH);
63     digitalWrite(rightMotorForwardPin, LOW);
64     digitalWrite(rightMotorBackwardPin, HIGH);
65 }
66 void stopCar() {
67     digitalWrite(leftMotorForwardPin, LOW);
68     digitalWrite(leftMotorBackwardPin, LOW);
69     digitalWrite(rightMotorForwardPin, LOW);
70     digitalWrite(rightMotorBackwardPin, LOW);
71 }
72 void goRight() {
73     digitalWrite(leftMotorForwardPin, HIGH);
74     digitalWrite(leftMotorBackwardPin, LOW);
75     digitalWrite(rightMotorForwardPin, LOW);
76     digitalWrite(rightMotorBackwardPin, LOW);
77 }
78 void goLeft() {
79     digitalWrite(leftMotorForwardPin, LOW);
80     digitalWrite(leftMotorBackwardPin, LOW);
81     digitalWrite(rightMotorForwardPin, HIGH);
82     digitalWrite(rightMotorBackwardPin, LOW);
83 }

```

Slika 4.12. Naredbe za kretanje i postavke nožica motora.

#### 4.4.3. Programski kod za drugu inačicu prototipa autonomnog vozila

Programski kod druge inačice nadogradnja je prve s razvijenijom logikom. Koristi se biblioteka `<Newping.h>` koja je sposobna za točno dohvaćanje vrijednosti s više senzora HC-SR04 te se naredbom `#include` uključuje u program. Naredbom `#define` definira se broj senzora koji je u ovome radu postavljen na tri, maksimalna udaljenost senzora na 200 cm i interval *pingova* na 33 (ms između svakog *pinga* senzora). Kreiraju se polja `pingTimer[]` koji služi za pohranu vremena kada svaki senzor daje *ping*, `sensors[]` za pohranjivanje udaljenosti senzora u centimetrima i `sensor[]` u kojemu se inicijaliziraju senzori. Polje `distance[]` služi za testiranje sposobnosti senzora za prepoznavanje objekta unutar vidnog polja. Postavljaju se i zastavice `OOR_flag` koje služe za kontroliranje je li senzor izvan dometa (*eng. out of range*). Treba naglasiti da se kroz testiranja utvrdilo da se senzori, ukoliko prijeđu maksimalnu vrijednost od 200 cm, postavljaju na nulu što će imati veliku ulogu u samim *if* izrazima. Nožice motora jednake su kao u prošloj inačici programskog koda. Slika 4.13 prikazuje gore navedeno.

```

1  #include <NewPing.h>
2
3  #define SONAR_NUM 3
4  #define MAX_DISTANCE 200
5  #define PING_INTERVAL 33
6  unsigned long pingTimer[SONAR_NUM]; // When each pings.
7  unsigned int cm[SONAR_NUM]; // Store ping distances.
8  uint8_t currentSensor = 0; // Which sensor is active.
9
10 long sensors[3];
11 unsigned long distance[3];
12
13 const unsigned long timeout = 200*58;
14 boolean OOR_flag_0 = false;
15 boolean OOR_flag_1 = false;
16 boolean OOR_flag_2 = false;
17
18 // left motor
19 int leftMotorSpeedPin = 3;
20 int leftMotorForwardPin = 3;
21 int leftMotorBackwardPin = 5;
22 // right motor
23 int rightMotorSpeedPin = 11;
24 int rightMotorForwardPin = 11;
25 int rightMotorBackwardPin = 12;
26
27 NewPing sonar[SONAR_NUM] = { // Sensor object array.
28   NewPing(A0, A1, MAX_DISTANCE), //frontSensor
29   NewPing(A2, A3, MAX_DISTANCE), //leftSensor
30   NewPing(A4, A5, MAX_DISTANCE), //rightSensor
31 };

```

Slika 4.13. Inicijalizacija biblioteke, senzora i potrebnih varijabli i polja.

Nakon inicijalizacije, definira se funkcija *sensorCycle()* koja dohvaća udaljenost izmjerenu od strane senzora. Naredba *millis()* vraća broj ms koje su prošle od trenutka kada sklop Croduino Basic3 počinje prolaziti kroz program te ukoliko je taj broj veći od *pinga* pojedinog senzora, *pingTimer* postavlja vrijednost na zbroj *ping* intervala pomnoženih s brojem senzora. Tada započinje funkcija *oneSensorCycle()* tj. glavni dio programa koji se bavi procesiranjem podataka i određivanjem idućih akcija prototipa autonomnog vozila. Nakon toga, zaustavlja se odbrojavanje trenutnog senzora, vrijednosti *cm[]* polja postavljaju se na nula i provjerava se odjek (*eng. echo*) *pinga* trenutnog senzora. Funkcija *echoCheck()* provjerava dolazi li do *pinga*, ako da, postavi udaljenost u polje *cm[]*. Slika 4.14 prikazuje metodu dohvaćanja udaljenosti od senzora.

```

33 void sensorCycle () { //getting the distance from sensors
34   for (uint8_t i = 0; i < SONAR_NUM; i++) {
35     if (millis() >= pingTimer[i]) {
36       pingTimer[i] += PING_INTERVAL * SONAR_NUM;
37       if (i == 0 && currentSensor == SONAR_NUM - 1) {
38         oneSensorCycle(); // Do something with results.
39       }
40       sonar[currentSensor].timer_stop();
41       currentSensor = i;
42       cm[currentSensor] = 0;
43       sonar[currentSensor].ping_timer(echoCheck);
44     }
45   }
46 }
47
48 void echoCheck() { // If ping echo, set distance to array.
49   if (sonar[currentSensor].check_timer())
50     cm[currentSensor] = sonar[currentSensor].ping_result / US_ROUNDTRIP_CM;
51 }

```

Slika 4.14. Metoda dohvaćanja udaljenosti od senzora.

Funkcija `oneSensorCycle()` najvažnija je funkcija programskog koda druge inačice prototipa autonomnog vozila jer se bavi logikom upravljanja autonomnog vozila. Započinje `for` petljom i provjerom udaljenosti senzora u centimetrima a funkcionalnost petlje prati se preko sučelja Arduino IDE i njegova Serial Monitora. Zatim se provjeravaju `if` izrazi da se utvrdi jesu li senzori izvan dometa, što je prikazano na slici 4.15. Ukoliko je jedan od senzora jednak nuli, dolazi do provjere o kojem se senzoru radi i zastavica senzora koji se nalazi izvan dometa postavlja se na `true`, dok u suprotnom zastavica ostaje `false`.

```

53 void oneSensorCycle() {
54   for (uint8_t i = 0; i < SONAR_NUM; i++) {
55     delay(20); //Wait 20 ms between pings.
56     sensors[i] = sonar[i].ping_cm();
57     if(i==0)distance[0] = round(pulseIn(A1, HIGH, timeout) /58);//Distance in CM's, use /148 for inches.
58     else if(i==1) distance[1] = round(pulseIn(A3, HIGH, timeout) /58);//Distance in CM's, use /148 for inches.
59     else distance[2] = round(pulseIn(A5, HIGH, timeout) /58);//Distance in CM's, use /148 for inches.
60   }
61
62   Serial.println("s0: ");
63   Serial.println(sensors[0]);
64   Serial.println("s1: ");
65   Serial.println(sensors[1]);
66   Serial.println("s2: ");
67   Serial.println(sensors[2]);
68   if (distance[0] == 0 || distance[1] == 0 || distance[2] == 0){//Reached timeout
69     if(distance[0] == 0){
70       OOR_flag_0 = true;
71     }else OOR_flag_0 = false;
72     if(distance[1] == 0){
73       OOR_flag_1 = true;
74     }else OOR_flag_1 = false;
75     if(distance[2] == 0){
76       OOR_flag_2 = true;
77     }else OOR_flag_0 = false;
78   }

```

Slika 4.15. Provjera dometa senzora.

Nadalje, inicijalizira se zastavica `back_flag` tipa `boolean`, tj. zastavica koja će odrediti kada se vozilo treba kretati unatrag. Slijedi niz `if` izraza koji provjeravaju slučaj za koji je zastavica istinita. Ukoliko je udaljenost prvog senzora nula i zastavica za provjeru prvog senzora izvan dometa istinita, nastavi kretanje prema naprijed jer se objekt nalazi izvan dometa. U suprotnom zaustavi vozilo, provjeri jesu li lijevi i desni senzori manji od 25. Ukoliko jesu, ako je prikazana nula, provjeri prikazuju li oba senzora je li objekt izvan dometa. U tom slučaju stanje zastavice se mijenja i vozilo se ne pomiče unatrag naredbom `goBackward()`. U suprotnom, vozilo se pomjera unatrag. Također, ukoliko je vrijednost lijevog senzora manja ili jednaka 25 provjerava se je li nula te ako je, provjerava se je li objekt izvan dometa, ako jest, poziva se naredba `goLeft()` i vozilo skreće lijevo. U suprotnom dolazi do provjere udaljenosti desnog senzora te ukoliko je ona veća od 25 ili se objekt nalazi izvan dometa senzora, dolazi do skretanja nadesno. Slika 4.16 prikazuje gore objašnjeno dok slika 4.17 prikazuje usporedbu lijevog i desnog senzora kada se objekt nalazi na udaljenosti većoj od 25 te se vozilo kreće u smjeru najveće udaljenosti.

```

79   boolean back_flag = false;
80   if(sensors[0] <= 25){
81       if(sensors[0] == 0 && OOR_flag_0 == true){goForward();}
82       delay(350);
83       if(sensors[1] <= 25 && sensors[2] <= 25) {
84           if(sensors[1] == 0 && OOR_flag_1 == true){
85               if(sensors[2] == 0 && OOR_flag_2 == true){
86                   back_flag = true;
87               }
88           }
89           if(!back_flag){
90               Serial.println("back");
91               goBackward();
92               delay(700);
93           }
94       }
95       else if(sensors[1] <= 25){
96           if(sensors[1] == 0 && OOR_flag_1 == true){goLeft();}
97           else if(sensors[2] > 25 || (sensors[2] == 0 && OOR_flag_2 == true)){
98               Serial.print("s1 - left: ");
99               Serial.println(sensors[1]);
100              goRight();
101          }
102      }
103      else if(sensors[2] <= 25){
104          if(sensors[2] == 0 && OOR_flag_2 == true){goRight();}
105          else if (sensors[1] > 25 || (sensors[1] == 0 && OOR_flag_1 == true)){
106              Serial.print("s2 - right: ");
107              Serial.println(sensors[2]);
108              goLeft();
109          }
110      }

```

Slika 4.16. Prvi dio koda za upravljanje kretanja vozilom.

```

111   }
112   if(sensors[1] > 25 && sensors[2] > 25){
113       if(sensors[1] > sensors[2]){
114           Serial.print("going left");
115           goLeft();
116       }
117       else {
118           goRight();
119       }
120   }
121   else{
122       Serial.println("forward");
123       goForward();
124   }
125 }
126

```

Slika 4.17. Drugi dio koda za upravljanje kretanja vozilom.

Unutar funkcije *setup()* u drugoj inačici programskog koda, jedini dodatak je kod prikazan na slici 4.18. Kod služi za pokretanje *pinga* te je namješten da se prvi *ping* pokreće nakon 75 ms te svaki sljedeći nakon 33 ms.

```

Serial.begin(115200);
pingTimer[0] = millis() + 75; // First ping start in ms.
for (uint8_t i = 1; i < SONAR_NUM; i++){
    pingTimer[i] = pingTimer[i - 1] + PING_INTERVAL;
}

```

Slika 4.18. Prikaz pokretanja pingova.

#### 4.4.4. Programski kod za treću inačicu prototipa autonomnog vozila

Za treću inačicu prototipa autonomnog vozila s kamerom CMOS OV7670, cilj je bio ostvarivanje povezanosti kamere autonomnog vozila s računalom sa sposobnošću dohvaćanja slike. Programski kod [43] korišten za povezivanje kamere s mikroupravljačem nije djelo autora ovoga rada te neće biti uključeno u sklopu završnoga rada. Sva prava na kod pripadaju njegovu pravom autoru.

#### 4.4.5. Komunikacija sustava autonomnog vozila

Komunikacija sustava autonomnog vozila odvija se preko mikroupravljača Croduino Basic3. Na sami pritisak okidača za paljenje motora, mikroupravljač Croduino Basic3 dobiva informaciju o pripravnosti motora i izvršava početnu funkciju *sensorCycle()* za dohvaćanje udaljenosti očitane sa senzora. Nakon prikupljanja tih informacija, ulazi u funkciju *oneSensorCycle()* predviđen za obradu podataka. Vrijednosti udaljenosti sa senzora pohranjuju se u niz *sensors[]* te je svakom senzoru predana pozicija,  $i = 0$  za prednji senzor,  $i = 1$  za lijevi senzor i  $i = 2$  za desni senzor. Tada mikroupravljač obrađuje podatke senzora i ukoliko prema očitanim vrijednostima procjenjuje da su vrijednosti prednjeg senzora manje od dozvoljenih između prototipa autonomnog vozila i objekta te da se ne može kretati prema naprijed naredbom *goForward()* koja je zadana kao prva akcija, poziva naredbu *goLeft()* za skretanje lijevo ukoliko je taj smjer najidealniji, naredbu *goRight()* za skretanje desno ukoliko se ne može kretati lijevo ili nastaviti naprijed, ili *goBackward()* za kretanje unatrag ukoliko su očitavanja sa svih senzora manja od dozvoljene udaljenosti između prototipa autonomnog vozila i objekata ispred njega. Sustav se nalazi u petlji gdje su očitavanja udaljenosti vrijednosti konstantna te sukladno uvjetima na stazi dolazi do procjene nastavka smjera kretanja. Kada vozilo procjeni da se može nastaviti kretati ravno, poziva se naredba *goForward()*. Sukladno određivanju naredbe koju će pozvati, naredba se šalje preko mikroupravljača Croduino Basic3 prema motorima koji tada izvršavaju određenu naredbu.

## **5. OPIS NAČINA KORIŠTENJA I ISPITIVANJE PROTOTIPA AUTONOMNOG VOZILA**

Nakon izgradnje sklopovskog i implementacije programskog dijela prototipa, u ovom poglavlju opisan je način korištenja prototipa autonomnog vozila. Nakon toga, prikazani su i analizirani rezultati ispitivanja prototipa autonomnog vozila, kao i primijenjeni postupak strojnog učenja zasnovan na algoritmu stabla odluke.

### **5.1. Opis korištenja prototipa autonomnog vozila**

Korištenje prototipa autonomnog vozila je iz korisničke perspektive jednostavno. Jedino što se treba napraviti je pritisnuti prekidač koji se nalazi na modelu prototipa autonomnog vozila kako bi se upalilo, i ostalo je sve na samome prototipu autonomnog vozila. Vozilo se kreće kroz prostor te ne ovisi o samome korisniku.

### **5.2. Uvjeti ispitivanja i slučajevi korištenja**

#### **5.2.1. Ispitni slučaj 1**

Prvi ispitni slučaj odvija se na prvoj inačici prototipa autonomnog vozila. Ispred modela su postavljene prepreke u obliku kutija, a na vozilu je da ih, ako je to moguće, izbjegne. Prva vožnja pokazuje se uspješnom, ukoliko je razmak između prepreka dovoljno velik za uočavanje. Uočeno je da ukoliko vozilo napravi skretanje u bilo kojem smjeru, a odmah nakon skretanja se ispred njega nalazi nova prepreka, vozilo nastavlja ići dalje i direktno se zabija u prepreku. Vrijeme dohvaćanja nove informacije o nenadanoj prepreci je presporo i vozilo ne reagira u željenom ishodu. Prvi ispitni slučaj pokazuje grešku u logici.

#### **5.2.2. Ispitni slučaj 2**

Preispitivanjem logike nakon prvog slučaja, dolazi do više neuspješnih inačica istog programa, do finalnog usavršavanja koda. Idući ispitni slučaj nalazi se u istim uvjetima kao i prvi te je autonomno vozilo, nakon početnog kretanja naprijed i pokušaja skretanja nadesno, uočilo novu prepreku, kretalo se unatrag da stvori više prostora za iduću akciju, skrenulo nadesno gdje je procijenilo da ima dovoljno prostora za kretanje između dvaju prepreka i nastavilo naprijed. Drugi ispitni slučaj prikazan na slici 5.1 u potpunosti uspješan za dane prepreke.





*Slika 5.1. Model autonomnog vozila u trenutku skretanja nadesno kako bi izbjeglo prepreke.*

### **5.2.3. Ispitni slučaj 3**

Prva inačica prototipa autonomnog vozila suočava se s većim brojem prepreka. Broj prepreka na putu povećan je s tri na sedam, slučajno raspoređenih na podu, nekih bliže jedne drugima, neke udaljenije. Ispitni slučaj 3 je neuspješan jer u slučajevima manje udaljenosti između dvaju prepreka dolazi do sudara s preprekama. Razlog tome je logičan jer jedan ultrazvučni senzor HC-SR04 ima vidokrug  $30^\circ$  što model autonomnog vozila ograničava u shvaćanju okoline oko njega. Ispitni slučaj tri dovodi do zaključka potrebe implementacije više senzora.

### **5.2.4. Ispitni slučaj 4**

Druga inačica prototipa autonomnog vozila po prvi put se suočava sa svojom okolinom. Prvobitne kutije koje su bile raspoređene po podu zamijenjene su poligonom u obliku elipse dimenzija 2 m sa 1.8 m. Prvi ispitni slučaj druge inačice jest snalaženje u zatvorenom prostoru prije ispitivanja sa složenijim preprekama. Prototip autonomnog vozila uspješno upravlja danom stazom, s rijetkim slučajevima kada se odluči sudariti sa zidom. Razlog sudara pripisuje se nemogućnosti smanjenja brzine motora zbog tehničkih problema s upravljačkim modulom L298N, što rezultira s obradom podataka koja nije dovoljno brza u stvarnom vremenu. Prototip se uspješno snalazi na poligonu s mogućnošću sudara u zid. Slika 5.2 prikazuje drugu inačicu prototipa autonomnog vozila za vrijeme ispitivanja.



*Slika 5.2. Druga inačica modela autonomnog vozila u trenutku uspješnog zaobilaženja zida.*

### **5.2.5. Ispitni slučaj 5**

Druga inačica prototipa autonomnog vozila susreće se sa slučajno raspoređenim preprekama na stazi. Vožnja između prepreka je uspješna s jednakim problemom kao i u prethodnom ispitnom slučaju, jedini novi uočeni problem jest kretanje unatrag gdje autonomno vozilo može srušiti prepreku ukoliko se nalazi preblizu nje same. Nadogradnja za taj problem je dodatni senzor koji bi pratio udaljenost modela autonomnog vozila od objekata koji se nalaze iza njega. S obzirom na to da autor nije u mogućnosti dodati još jedan senzor, stražnji senzor HC-SR04 je ideja za nadogradnju modela, čija implementacija u sam projekt nije teška. Za svrhu projekta, kretanje unatrag je smanjeno, što je samo privremeno rješenje, ali je za ispitni slučaj bilo dovoljno.

### **5.2.6. Ispitni slučaj 6**

Ispitni slučaj provjerava sposobnost povezivanja treće inačice prototipa autonomnog vozila opremljene kamerom CMOS OV7670 s računalom i kvalitetu slike. S obzirom na ograničenje USB kabela koji ih povezuje, provjera mobilnosti se ne odvija. Provjera povezivanja kamere s računalom prvi je korak implementiranja kamere u sklop modela autonomnog vozila, koja bi u idućim inačicama projekta služila kao glavno sredstvo dohvaćanja podataka okoline i omogućila naprednije strojno učenje s primjenom grupiranja k-srednjih vrijednosti, višerazredne neuronske mreže, Bayesovog algoritma, regresiju i klasifikaciju. Autor bi u tu svrhu, pored Croduina Basic3 ugradio u model i Raspberry Pi koji bi omogućavao bolju kontrolu okoliša u kojem se autonomno vozilo nalazi i obradu podataka kamere. Sensori bi dodatno osiguravali udaljenost između vozila i objekata, posebice u slučaju parkiranja vozila, ali bi se navigacija prostorom odvijala preko

kamere i strojnim učenjem koje bi bilo u stanju razaznati koji se točno objekt pred vozilom nalazi kao i navigaciju unutar traka za vožnju. Autor bi za tu inačicu promijenio i kameru na jednu kvalitetniju jer kamera CMOS OV7670 u sklopu naprednijeg modela ne zadovoljava uvjete. Ideja implementacije složenijeg strojnog učenja morala se odbaciti jer kamera CMOS OV7670 nije u stanju prenositi uživo video zapis, jedino fotografirati. Ispitni slučaj 6 uspješno je odrađen.

### **5.3. Analiza primijenjenog postupka strojnog učenja**

Jednostavan model stabla odluke implementiran u model autonomnog vozila pokazao se uspješan u granicama ovoga rada, iako poprilično ograničenog sklopovskim dijelom. Druga inačica prototipa autonomnog vozila u koju je implementiran, zahtijevala je obradu trostrukog dohvaćanja podataka s triju senzora HC-SR04 i njihovu brzu obradu za donošenje odluka. Algoritam stabla odluke neparametarska je diskriminantna metoda koja se u ovome projektu koristi kao klasifikator, odnosno klasu čine smjerovi kretanja. Uzorci podataka prikupljeni sa senzora i pohranjeni u polja, predani su u strukturu stabla gdje se uspoređuje vrijednost udaljenosti i prema njoj dolazi do pravilne odluke kretanja. Algoritam, iako jednostavan za razliku od njegovih sofisticiranih inačica, predstavlja logičke ispitne slučajeve u kojemu se autonomno vozilo nalazi i izlaze tj. u slučaju rada naredbe, predane ovisno o ulaznim vrijednostima udaljenosti sa senzora. Algoritam jednostavnog stabla odluke logički i strukturno je točan, iako ponekad zbog manjkavosti kvalitete senzora i njihove ograničenosti, zna se dogoditi da algoritam kasni s naredbom. Točnost algoritma prikazana je u tablici 5.1 nakon deset ispitivanja od kojih je svako trajalo jednu minutu. Točnost stabla odluke dobivena je dijeljenjem broja sudara, odnosno krivim odlukama, s ukupnim brojem napravljenih odluka, odnosno kretanja. Iako je u teoriji algoritam točan, u praktičnoj primjeni na njega utječu druge komponente modela autonomnog vozila. S obzirom na to da je stablo odluke primijenjeno u radu jednostavan model strojnog učenja, njegova točnost je zadovoljavajuća.

### **5.4. Rezultati ispitivanja prototipa autonomnog vozila**

Provedeno je deset ispitivanja za prvu i drugu inačicu modela autonomnog vozila. Obje verzije prošle su isti poligon s jednakim pozicijama prepreka unutar poligona. Za svaku vožnju trajanja od jedne minute, praćen je broj sudara sa zidovima poligona i ukupan broj odluka kretanja provedenih unutar jedne minute. Prema rezultatima određen je prosječan broj sudara unutar jedne minute i točnost algoritma stabla odluke. Svi rezultati ispitivanja prikazani su u tablici 5.1.

Tablica 5.1. Skupni rezultati ispitivanja prototipa autonomnog vozila

<i>Ispitna vožnja</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>Prosjek sudara u min</i>	<i>Prosječan broj odluka</i>	<i>Točnost stabla odluke</i>
<i>Broj sudara bez prepreka na poligonu prve inačice modela</i>	6	8	3	5	6	5	7	4	6	7	5.7	-	-
<i>Broj napravljenih odluka tijekom vožnje</i>	49	42	59	45	42	46	41	55	44	47	-	47	0.8787
<i>Broj sudara s preprekama na poligonu prve inačice modela</i>	7	13	10	15	11	6	8	8	7	10	9.5	-	-
<i>Broj napravljenih odluka tijekom vožnje</i>	53	47	51	50	46	44	52	55	46	40	-	48.4	0.8037
<i>Broj sudara bez prepreka na poligonu druge inačice modela</i>	4	3	6	4	5	8	4	5	6	3	4.8	-	-
<i>Broj napravljenih odluka tijekom vožnje</i>	78	76	68	73	82	66	71	83	69	74	-	74	0.9351
<i>Broj sudara s preprekama na poligonu druge inačice modela</i>	6	5	5	7	5	6	5	6	7	5	5.7	-	-
<i>Broj napravljenih odluka tijekom vožnje</i>	88	96	102	91	85	79	93	96	88	103	-	92.1	0.9381

Ispitivanja su pokazala različite rezultate kroz razvoj prototipa autonomnog vozila. Rezultati pokazuju da je druga inačica modela autonomnog vozila uspješnija i algoritam točniji negoli prve

inačice. U oba slučaja ispitivanja na poligonu druga inačica pokazuje točnost iznad 90 posto, dok prva inačica iznad 80 posto. Ova uspješnost pripisuje se nadogradnjama i usavršavanju prototipa prema prvih šest ispitnih slučajeva. U ispitnom slučaju 1 prvi put je ukazano na pogrešku u logici, ispitni slučaj 3 ukazuje na potrebu više senzora dok ispitni slučaj 5 ukazuje na potrebu stražnjeg senzora HC-SR04. Svaka iduća nadogradnja nakon ukazanih pogrešaka rezultira s uspješnošću s rijetkim pogreškama koje nastaju kao rezultat ograničenosti dometa, vidokruga ili komponenti. Jedan od nedostataka mobilne platforme jest pozicioniranje senzora HC-SR04, tj. nedovoljno povećanje vidokruga autonomnog vozila. Naime, da je pozicija lijevog i desnog senzora HC-SR04 pod kutom 90°, očitavanja i reakcije autonomnog vozila bile bi još preciznije, a pogreške smanjene na minimum. Daljnjim ispitivanjima dobilo bi se još informacija i podataka za daljnje usavršavanje prototipa autonomnog vozila, iako kvalitetniji senzori i kamera bi omogućili sofisticiraniji razvoj prototipa autonomnog vozila.

## ZAKLJUČAK

Cilj ovog završnog rada bio je izrada prototipa autonomnog vozila sa sposobnošću izbjegavanja prepreka. U radu su opisani i analizirani sklopovski i programski zahtjevi na model autonomnog vozila kroz dvije inačice, predloženo sklopovsko i programsko rješenje za obradu podataka sa senzora koji podrazumijeva primjenu nadziranog strojnog učenja, odnosno klasifikacije s ciljem predviđanja na temelju implementiranog algoritma stabla odluke. Učinkovitost i točnost tog algoritma analizirani su kroz veći broj ispitivanja. Sklopovsko i programsko rješenje nadograđivano je kroz dvije inačice, a predložena je ideja za treću inačicu. Prva inačica vozila s jednim senzorom omogućuje jednostavno kretanja u prostoru i ima sposobnost uočavanja predmeta ispred sebe ali ima i problem pri predstavljanju prepreka u poligonu što smanjuje učinkovitost ove inačice. Druga inačica opremljena s tri senzora u koju je implementiran algoritam stabla odluke omogućuje veći vidokrug modela autonomnog vozila. Ona se uspješno kreće poligonom neovisno o preprekama koje obilazi u stvarnom vremenu ali ne isključuje poneki sudar. Treća inačica modela prototipa autonomnog vozila predlaže ali u ovom radu ne rješava nadogradnju druge inačice implementiranjem drugih učinkovitijih algoritama strojnog učenja kao što su grupiranje algoritmom *k-means*, višerazredne neuronske mreže, Bayesovog algoritma, i drugih.

Prototip autonomnog vozila ima mnogo prostora za poboljšanje. Primjenom kvalitetnijih senzora, kamere i snažnijeg mikroupravljača, prototip autonomnog vozila bio bi u stanju kretati se cestovnim trakom, samostalno parkirati, raspoznavati prometne znakove i semafore, te raspoznavati ljude, životinje ili nežive objekte na cesti.

## LITERATURA

- [1] J. Contreras-Castillo, J. Guerrero-Ibáñez i S. Zeadally, »Autonomous Cars: Challenges and Opportunities,« *IT Professional*, svez. 21, br. 6, pp. 6-13, 2019.
- [2] M. Weber, »Where To? A History of Autonomous Vehicles,« Computer History Museum, 8 Svibanj 2014. <https://computerhistory.org/blog/where-to-a-history-of-autonomous-vehicles/?key=where-to-a-history-of-autonomous-vehicles>. (11. kolovoz 2021.)
- [3] R. E. Fenton i R. J. Mayhan, »Automated Highway Studies at the Ohio State University-An Overview,« *IEEE Transactions on Vehicular Technology*, svez. 40, br. 1, pp. 100-113, 1991.
- [4] I. Petros, *Automated Highway Systems*, Springer, 2013.
- [5] J. Schmidhuber, »Prof. Schmidhuber's Highlights of Robot Car History.,« 2009. <https://people.idsia.ch/~juergen/robotcars.html>. (11. kolovoz 2021.)
- [6] J. W. Lowrie, M. Thomas, K. Gremban i M. Turk, »The Autonomous Land Vehicle (ALV) Preliminary Road-Following Demonstration,« u *Proc. SPIE*, 1985.
- [7] C. Lee, »This Self-driving Car Drove Safely All Over South Korea – in 1993,« *Techwire Asia*, 3. travanj 2021. <https://techwireasia.com/2021/04/this-self-driving-car-drove-safely-all-over-south-korea-in-1993/>. (11. kolovoz 2021.)
- [8] P. Szikora i N. Madarász, »Self-driving Cars — The Human Side,« u *2017 IEEE 14th International Scientific Conference on Informatics*, 2017.
- [9] ACEA, »The Automobile Industry Pocket Guide 2020-2021,« ACEA, 30 Srpanj 2020. <https://www.acea.auto/publication/automobile-industry-pocket-guide-2020-2021/>. (11.kolovoz 2021.)
- [10] »Waymo Project, Waymo, 2018.,« 2018. <https://waymo.com.4..> (7. lipanj 2021.)
- [11] »Tesla Autopilot,« Tesla, 2018. [https://www.tesla.com/es\\_MX/autopilot](https://www.tesla.com/es_MX/autopilot). (7. lipanj 2021.)
- [12] J. Markoff, »Google Cars Drive Themselves, in Traffic,« *The New York Times*, 2010. <https://www.nytimes.com/2010/10/10/science/10google.html>. (15. srpanj 2021.)
- [13] WHO, “Road traffic injuries,” 21. lipanj 2021. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>. (7. srpanj 2021.)
- [14] S. Singh, »Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey,« 2015. <https://crashstats.nhtsa.dot.gov/>. (7. srpanj 2021.)

- [15] N. Sousa, J. Coutinho-Rodrigues, A. Almeida i E. Natividade-Jesus, »Dawn of Autonomous Vehicles: Review and Challenges Ahead,« *Proceedings of the Institution of Civil Engineers - Municipal Engineer*, svez. 171, br. 1, pp. 3-14, 2018.
- [16] S. A. Bagloee, M. Tavana, M. Asadi i T. Oliver, »Autonomous Vehicles: Challenges, Opportunities, and Future Implications for Transportation Policies,« *Journal of Modern Transportation*, svez. 24, br. 4, pp. 284-303, 2016.
- [17] L. Blain, »Self-driving Vehicles: What Are the Six Levels of Autonomy?,« 2017. <http://newatlas.com/sae-autonomous-levels-definition-self-driving/49947/>. (7. srpanj 2021.)
- [18] Á. Takács, I. Rudas, D. Bösl i T. Haidegger, »Highly Automated Vehicles and Self-Driving Cars,« *IEEE Robotics & Automation Magazine.*, svez. 25, br. 4, pp. 106-112, 2018.
- [19] H. Link, C. Nash, A. Ricci i J. Shires, »A Generalised Approach for Measuring the Marginal Social Costs of Road Transport in Europe,« *International Journal of Sustainable Transportation*, svez. 10, br. 2, pp. 105-119, 2014.
- [20] I. Parry, M. Walls i W. Harrington, »Automobile Externalities and Policies,« *Journal of Economic Literature*, svez. 45, br. 2, pp. 373-399, 2007.
- [21] J. M. Anderson, N. Kalra, K. D. Stanley, P. Sorensen, C. Samaras i O. T. A., »Autonomous Vehicle Technology: A Guide for Policy Makers,« *Rand Corporation*, 2016.
- [22] T. Litman, *Autonomous Vehicle Implementation Predictions - Implications for Transport Planning*, Victoria Transport Policy Institute, 2021.
- [23] W. J. C. Melis, »The Future of Electrical Vehicles,« *Autonomous Vehicles: Intelligent Transport Systems and Smart Technologies*, p. 509–530, 2014.
- [24] N. Kang, F. M. Feinberg i P. Y. Papalambros, »Autonomous Electric Vehicle Sharing System Design,« u *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Boston, 2015.
- [25] S. Palvia i V. Vemuri, »Forecasts of Jobless Growth: Facts and Myths,« *Journal of Information Technology Case and Application Research*, svez. 18, br. 1, pp. 4-10, 2016.
- [26] A. Greenberg, »Hackers Remotely Kill a Jeep on the Highway—With Me in It,« WIRED, 21. srpanj 2015. <https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>. (12. kolovoz 2021.)



- [27] O. Solon, »Team of Hackers Take Remote Control of Tesla Model S from 12 Miles Away,« The Guardian, 20. rujan 2016. <https://www.theguardian.com/technology/2016/sep/20/tesla-model-s-chinese-hack-remote-control-brakes>. (12. kolovoz 2021.)
- [28] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao i K. Zieba, »End to End Learning for Self-Driving Cars,« 25. travanj 2016. <https://arxiv.org/pdf/1604.07316.pdf>. (13. kolovoz 2021.)
- [29] J. D'Allegro, »How Google's Self-Driving Car Will Change Everything,« Investopedia, 20. lipanj 2021. <https://www.investopedia.com/articles/investing/052014/how-googles-selfdriving-car-will-change-everything.asp>. (15. kolovoz 2021.)
- [30] E. Guizzo, »How Google's Self-driving Car Works,« IEEE Spectrum Online, 18. listopad 2011. <https://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>. (15. srpanj 2021.)
- [31] »Waymo Driver,« Waymo, <https://waymo.com/waymo-driver/>. (15. kolovoz 2021.)
- [32] D. Dolgov, »How We've Built the World's Most Experienced Urban Driver,« Waymo, 19. kolovoz 2021. <https://blog.waymo.com/2021/08/MostExperiencedUrbanDriver.html>. (16. kolovoz 2021.)
- [33] »Tesla Autopilot,« Tesla, <https://www.tesla.com/autopilot>. (17. kolovoz 2021.)
- [34] »Tesla,« Tesla, <https://www.tesla.com/>. 22. kolovoz 2021.)
- [35] R. Hussain i S. Zeadally, »Autonomous Cars: Research Results, Issues and Future Challenges,« *IEEE Communications Surveys & Tutorials*, svez. 21, br. 2, pp. 275-1313, 2019.
- [36] F. Munir, S. Azam, M. Hussain, A. Sheri i M. Jeon, »Autonomous Vehicle: The Architecture Aspect of Self Driving Car,« u *SSIP 2018: Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing.*, 2018.
- [37] A. Gupta, »Machine Learning Algorithms in Autonomous Driving,« IIoT World, 15. ožujak 2018. <https://iiot-world.com/machine-learning/machine-learning-algorithms-in-autonomous-driving/>. (14. rujan 2021.)
- [38] »1.10. Decision Trees,« Scikit Learn, <https://scikit-learn.org/stable/modules/tree.html>. (15. ujan 2021.)
- [39] Sparkfun, <https://www.sparkfun.com/products/15569>. (18. kolovoz 2021.)

- [40] »Croduino Basic3,« e-radionica, <https://e-radionica.com/hr/croduino-basic3.html>. (18. kolovoz 2021.)
- [41] »KKM: DC Motor Driver Dual H-Bridge,« e-radionica, <https://e-radionica.com/hr/blog/2017/12/10/kkm-dc-motor-driver-dual-h-bridge/>. (18. kolovoz 2021.)
- [42] »ArduCam,« [https://www.arducam.com/wp-content/uploads/2019/11/Arducam\\_ov7670\\_cmos\\_camera\\_module\\_revc\\_ds.pdf](https://www.arducam.com/wp-content/uploads/2019/11/Arducam_ov7670_cmos_camera_module_revc_ds.pdf). (18. kolovoz 2021.)
- [43] »OV7670 Camera and Arduino,« Lulu's blog, <https://lucidar.me/en/arduino/camera-ov7670-and-arduino/>. (5. rujan 2021.)

## SAŽETAK

U ovom završnom radu prikazan je i ostvaren razvoj modela prototipa autonomnog vozila sa sposobnošću samostalne vožnje u okolini s preprekama. Na početku su opisani problemi i izazovi današnjih autonomnih vozila s naglaskom na postupke usmjerene obilaženju prepreka. Nakon postavljenih zahtjeva i predloženog sklopovskog i programskog modela prototipa vozila kroz više inačica prikazane su mogućnosti takvog vozila. Dok je u prvoj inačici s jednim senzorom mogućnost obilaska prepreka niska, druga inačica ima tri senzora i mogućnost donošenja odluka o kretanju na temelju podataka o udaljenosti i to uz primjenu algoritma stabla odluke. Sklopovski dio prototipa autonomnog vozila temelji se na okolini Arduino, a konačni oblik prototipa ispitan je i analiziran za više scenarija kretanja vozila sa i bez prepreka, pri čemu je analizirana i točnost primijenjenog algoritma stabla odluke. Također, predložen je i treći oblik modela prototipa autonomnog vozila s proširenim mogućnostima.

**Ključne riječi:** algoritam stabla odluke, Arduino, prepreke, prototip autonomnog vozila, strojno učenje.

## **ABSTRACT**

In this final paper, “Assembly and coding of prototype of autonomous vehicle based on mobile platform and machine learning procedures”, the development of a prototype model of an autonomous vehicle with the ability to drive independently in an environment with obstacles is presented and achieved. At the beginning, the problems and challenges of today's autonomous vehicles are described with an emphasis on obstacle-avoiding procedures. After the set requirements and the proposed hardware and software model of a prototype vehicle, the possibilities of such a vehicle are presented through several versions. While in the first version with one sensor the possibility of bypassing obstacles is low, the second version has three sensors and the ability to make movement decisions based on distance data using the decision tree algorithm. The hardware part of the autonomous vehicle prototype is based on the Arduino environment, and the final shape of the prototype was tested and analysed for several scenarios of vehicle movement with and without obstacles, and the accuracy of the applied decision tree algorithm was analysed. Also, a third form of autonomous vehicle prototype model with extended capabilities has been proposed.

**Keywords:** decision tree algorithm, Arduino, obstacles, autonomous vehicle prototype, machine learning.

## ŽIVOTOPIS

Tea Krčmar rođena je 23. lipnja 1998. godine u Osijeku, Hrvatska. 2010. i 2011. godine sudjeluje na 5. i 6. ljetnom kampu tehnike u Belom Manastiru s robotikom kao odabranim smjerom, a interes za programiranjem nastavlja pohađanjem prirodoslovno-matematičke gimnazije u Osijeku sa smjerom napredne informatike kao izbornog predmeta. Za vrijeme srednjoškolskog obrazovanja sudjeluje subotama na okupljanju informatičara u svrhu rješavanja programskih zadataka i usavršavanje znanja. Nakon završetka gimnazije upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

---

Potpis autora

## **PRILOZI (na DVD-u)**

**Prilog 1:** Završni rad u docx i pdf formatu.

**Prilog 2:** Arduino studio kod za sklop Croduino Basic3

**Prilog 3:** Sheme spajanja pojedinih inačica prototipa