

ŠKOLA ZA GRAFIKU, DIZAJN I  
MEDIJSKU PRODUKCIJU

**ARDUINO AUTONOMNI  
ROBOT ZA PRAĆENJE  
LINIJE**

ZAVRŠNI RAD

Predmet: Programiranje za web

Usmjerenje: Web dizajner

Mentor: Mario Sever

Učenica: Klara Makek 4.c

Zagreb 2022.

# SADRŽAJ

1. UVOD	1
2. ARDUINO MIKROKONTROLER	2
2.1. ARDUINO SPECIFIKACIJE	2
2.2. ARDUINO BAZIČNI JEZIK	3
3. MILE	4
3.1. ELEKTRONIČKE KOMPONENTE ROBOTA	4
3.2. SPAJANJE KOMPONENTI	5
4. 3D PRINTANJE ŠASIJE	6
4.1. 3D	6
4.2. TINKERCAD	6
4.3. ENDER 3 PRO	8
5. PROGRAMIRANJE ROBOTA	10
5.1. PRETRAŽIVANJE DOKUMENTACIJE	10
5.2. INSTRUCTABLES CIRCUITS	10
5.3. FORBIDDENBIT.COM	12
5.4. PRVI KOD	13
5.4.1. PSEDUDO JEZIK	13
5.4.2. DIJAGRAM TIJEKA	14
5.4.3. PROGRAMSKI KOD	15
5.5. POPRAVAK KODA	19
5.6. TESTIRANJE	22
5.7. ZAVRŠNO USAVRŠAVANJE	22
6. ZAKLJUČAK	23
LITERATURA	24
POPIS SLIKA	25

## 1.UVOD

Svijet se vrlo brzo mijenja pa tako i mi sa njime. Svakodnevno tisuće robota i softverskih programa obavlja poslove koje su nekada obavljali ljudi. Tehnologija obuhvaća naš svakodnevni život i ne možemo bez nje. Konstantno se pitamo kako da ju poboljšamo, pojednostavimo ili upotrijebimo.

Za to su potrebne ideje, mladi umovi i različita razmišljanja. Danas je sve češće poticanje mladih ljudi u IT industriju uz pomoć natječaja i natjecanja. Stotine pojedinaca upravo tako dolaze do fantastičnih izuma. Ideju za svoj završni rad sam dobila gledajući radove na strukovnom natjecanju srednjih škola Worldskills Croatia.

U izradi sam koristila Arduino Uno mikrokontroler kao bazu, elektroničke komponente, senzore i C++ programski jezik. Robot kojeg sam napravila i nazvala „Mile” prati jednostavnu crnu liniju na bijeloj pozadini te se tako autonomno kreće po označenom poligonu.

Ideja završnog rada je da na jednostavan način simulira autonomne aute bez pomoći algoritma, te da ga svaka osoba može uz malo želje napraviti. Kasnije se može nadograđivati po želji korisnika.

## 2. ARDUINO MIKROKONTROLER

### 2.1. ARDUINO SPECIFIKACIJE

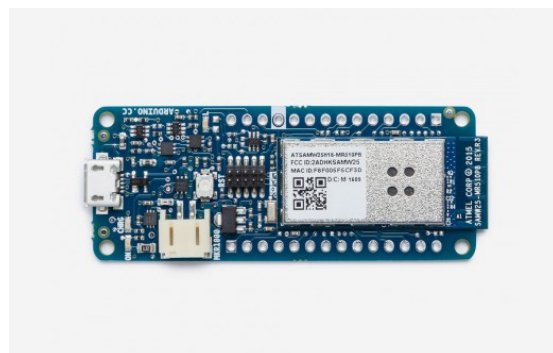
Arduino je tvrtka otvorenog kod za hardver i softver, projektna i korisnička zajednica koja dizajnira i proizvodi mikrokontrolere i setove za mikrokontroler za izgradnju digitalnih uređaja. Njegovi hardverski proizvodi imaju licencu CC-BY-SA, dok je softver licenciran pod GNU Lesser General Public License (LGPL) ili GNU General Public License (GPL), što omogućuje proizvodnju Arduino ploča i distribuciju softvera.

Dizajn ploče Arduino koristi razne mikroprocesore i kontrolere. Ploče su opremljene sklopovima pin-ova za digitalni i analogni ulaz / izlaz (1/0) koji se mogu povezati s raznim pločama za proširenje ('štitovi') ili pločama (za izradu prototipa) i drugim sklopovima. Ploče imaju serijska komunikacijska sučelja, uključujući Universal Serial Bus (USB) na nekim modelima, koja se također koriste za učitavanje programa s osobnih računala. Mikrokontroleri se mogu programirati pomoću programskih jezika C i C ++, koristeći standardni API koji je poznat i kao "Arduino jezik".

Uz upotrebu tradicionalnih alatnih lanaca za izgradnju, projekt Arduino nudi integrirano razvojno okruženje (IDE) i alat za naredbene retke (arduino-cli) razvijen u programu Go.



*Slika 1. Arduino logo*



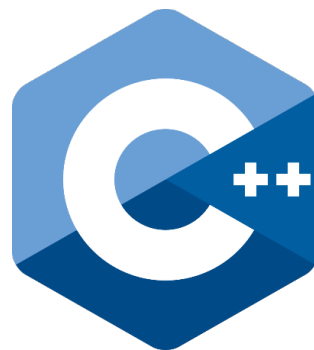
*Slika 2. Arduino mikrokontroler*

## 2.2. ARDUINO BAZIČNI JEZIK

Arduino je baziran na C i C++ jeziku, ali nit jedan od njih nije njegov glavni jezik. Njegov glavni jezik zove se Arduino. C++ se većinom koristi, ali C++ biblioteka neće Vam pomoći u programiranju. Arduino ima gotove biblioteke koje Vam pojednostave programiranje.



*Slika 3.  
C programski jezik*



*Slika 4.  
C ++ programski jezik*

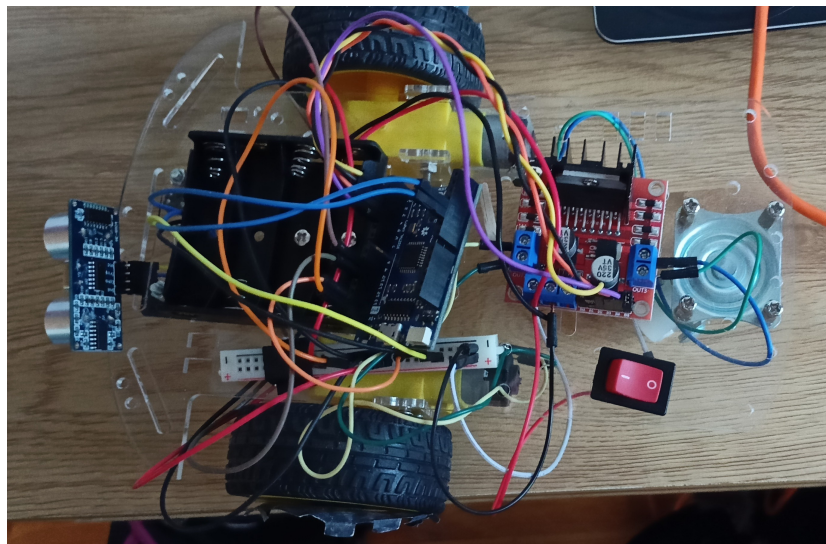
C ++ je programski jezik opće namjene koji je stvorio Bjarne Stroustrup kao proširenje programskog jezika C ili "C with Classes". Jezik se s vremenom značajno proširio, a moderni C ++ sada ima objektno orijentirane, generičke i funkcionalne značajke, kao i značajke upravljanja memorijom na niskoj razini. Gotovo se uvijek implementira kao kompajlirani jezik, a mnogi dobavljači nude kompajlere za C ++, uključujući Free Software Foundation, LLVM, Microsoft, Intel, Oracle i IBM, pa je dostupan na mnogim platformama.

### 3.MILE

#### 3.1. ELEKTRONIČKE KOMPONENTE ROBOTA

Potrebne komponente za izradu Mileta su se mijenjale kako je projekt trajao. Na početku sam se koristila sa: 1 metalnom šasijom, 1 Arduino MKR1000 , 27 muških kabela, eksperimentalna ploča, 1 čip za vožnju motora L293d, 2 Ir senzora za praćenje linije i komplet Smartcar 4wd u kojem dolaze 2 robotska kućišta, 6 L30 odstoynika, 4 dc Servo motora, 4 kotačića, kutija za baterije, 10 matica, 14 m3\*8 vijaka, 8 m3\*30 vijaka, 8 jezika i 4 encoded kotačića.

Nakon daljnjeg testiranja neke glavne komponente su se promijenile, a neke su kompletno izbačene radi prevelikog napora nad kotačima. Nove komponente se sastoje od: Arduino Uno, 1 3D izmodelirana šasija, 2 Ir senzora za praćenje linije, L293d štit pokretača motora za upravljanje motorom, 6 L30 odstoynika, 4 dc Servo motora, 4 kotačića, 10 matica, 14 m3\*8 vijaka, 8 m3\*30 vijaka, 8 jezika, 19 žica te 1 prekidač za uključivanje. (Slika 6.)



*Slika 5. Dijelovi Mileta*

### 3.2. SPAJANJE KOMPONENTI

Kroz 3D izmodelirano robotsko kućište smo provukli jezike kroz označene rupe na robotskom kućištu te Servo motore zaljepili za kućište. S druge strane smo zaljepili Arduino uno, L293d štit pokretača motora za upravljanje motorom, 2 Ir senzora za praćenje linije i 1 prekidač za uključivanje.

Stavimo L293d štit pokretača motora za upravljanje motorom na Arduino Uno. Na Ir senzor za praćenje priključimo žicu koja će biti kasnije spojena na GND, žicu za VCC/V5 i žicu za analogni pin te ponovimo to još jednom za drugi Ir senzor. Žice za GND spojimo skupa te dodamo dodatnu žicu koja će spajati te dvije žice u jednu i voditi na L293d štit. Ponovit ćemo to za VCC/V5 žice. Dvije žice koje smo stavili za analogne pinove ćemo staviti u A0(lijevi ir senzor) i A1(desni senzor) na L293d štitu.

Zatim ćemo uzeti dvije žice za svaki Servo motor i zalemiti ih na njih. Svaki servo motor bi trebao imati dvije žice, jednu za GND i drugu za VCC/V5. Te žice onda provučemo kroz označene rupe na šasiji i stavimo ih na: M1, M2, M3 i M4 označena mjesta na L293d štitu.

Za prekidač za uključivanje ćemo zalemiti dvije žice koje će služiti za GND i VCC/V5. Žice ćemo priključiti na označeno mjesto „POWER ->” na L293d štitu. No prije toga ćemo VCC/V5 žicu spojiti sa Vin žicom i nju priključiti u označeni Vin pin na L293d štitu.

## 4. 3D PRINTANJE ŠASIJA

### 4.1. 3D

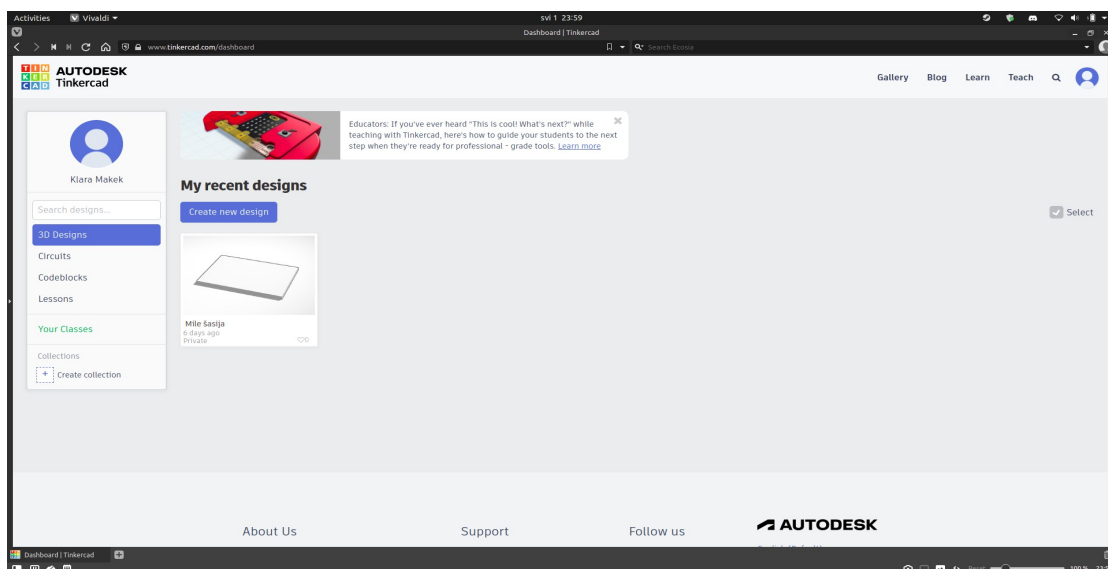
Trodimenzijski prostor (također: 3D prostor, 3-prostor ili, rijetko, trodimenzijski prostor) je geometrijska postavka u kojoj su tri vrijednosti (zване parametri) potrebne za određivanje položaja elementa (tj. točke). Ovo je neformalno značenje pojma dimenzija.

Koristeći se 3D modelacijom napravila sam šasiju za Mileta. Program Tinkercad kojeg sam koristila služi za jednostavno modeliranje.

### 4.2. TINKERCAD

Tinkercad je besplatna online zbirka softverskih alata koji pomažu ljudima diljem svijeta razmišljati, stvarati. Idealan uvod u Autodesk, lidera u 3D dizajnu, inženjeringu i softveru za zabavu.

Pomoću Tinkercada sam izmodelirala jednostavu šasiju za Mileta. Zadatak mi je bio napraviti što jednostavniju i fizički lakšu šasiju kako nebi bilo previše opterećenja na kotačima.

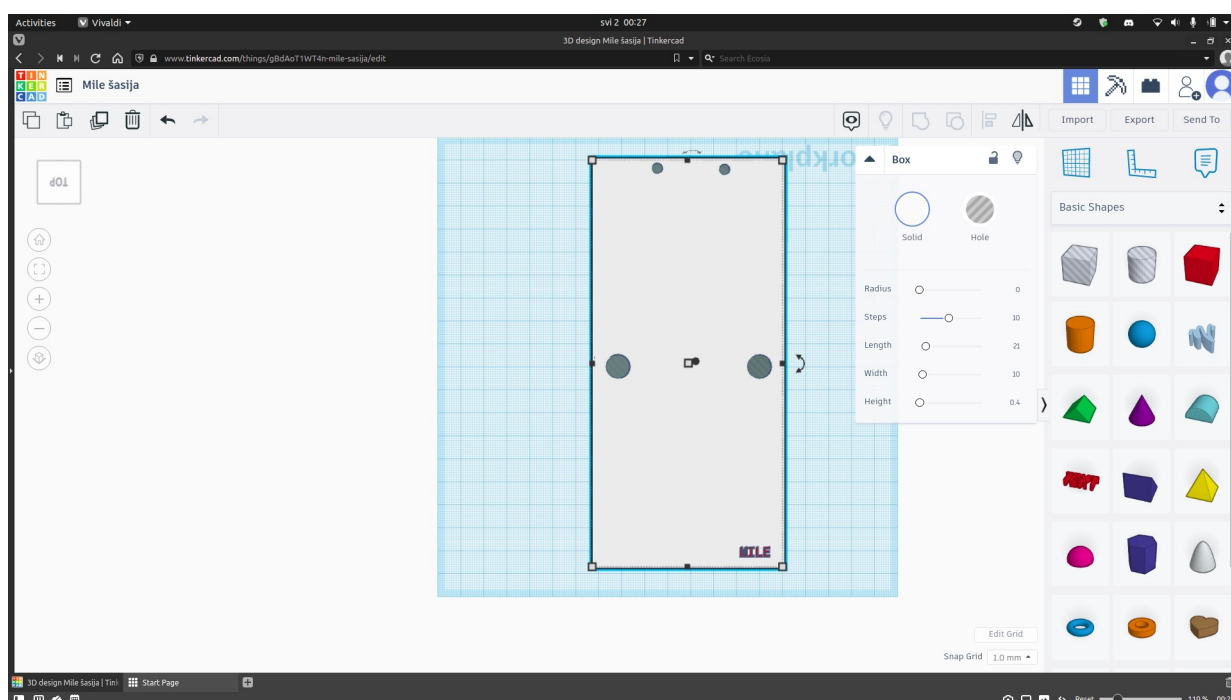


Slika 6. Tinkercad



Koristeći se već napravljenim oblikom kocke koju sam oblikovala po svojoj želji nastala je podloga. Podloga je debela 3.5mm, široka 10cm i dugačka 21cm. Mjere sam uzela tako da sam izmjerila staru metalnu šasiju i oduzela višak metala koji je visio sa strane.

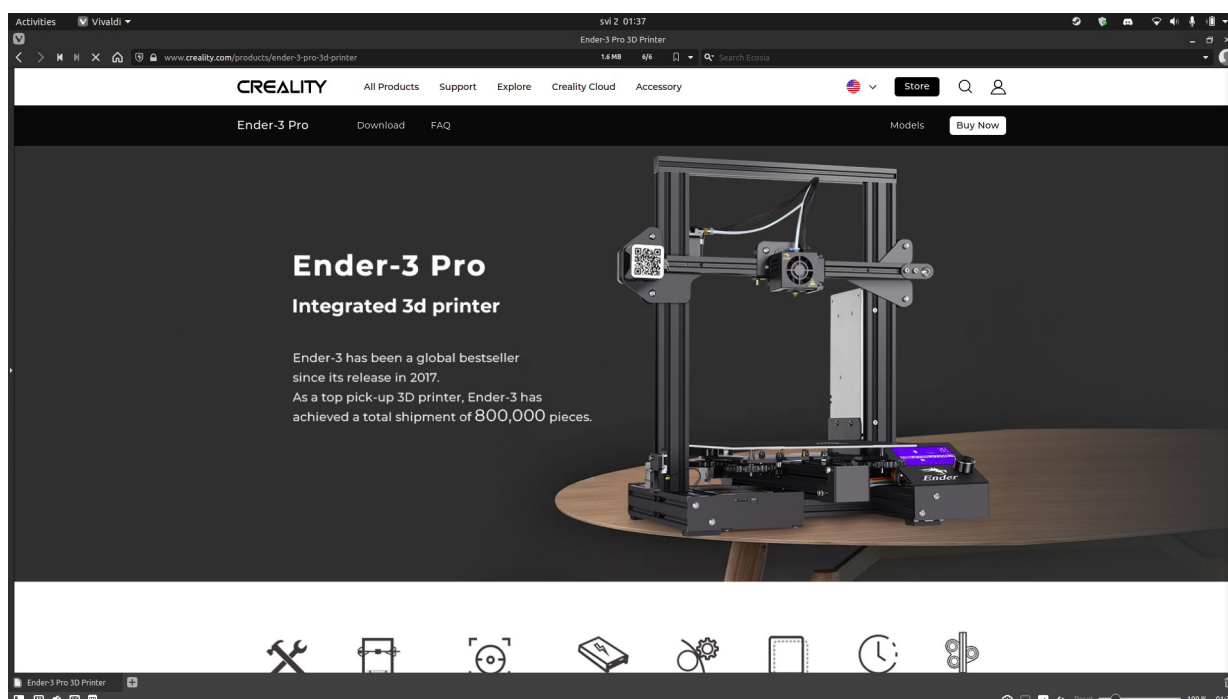
Zatim sam izbušila rupe na podlozi gdje bi žice od Ir senzora i Servo motora išle. Te na kraju potpisala Miletovo ime na dno podloge koristeći se izbočenim slovima.



*Slika 7. Dizajn šasije u Tinkercadu*

### 4.3. ENDER 3 PRO

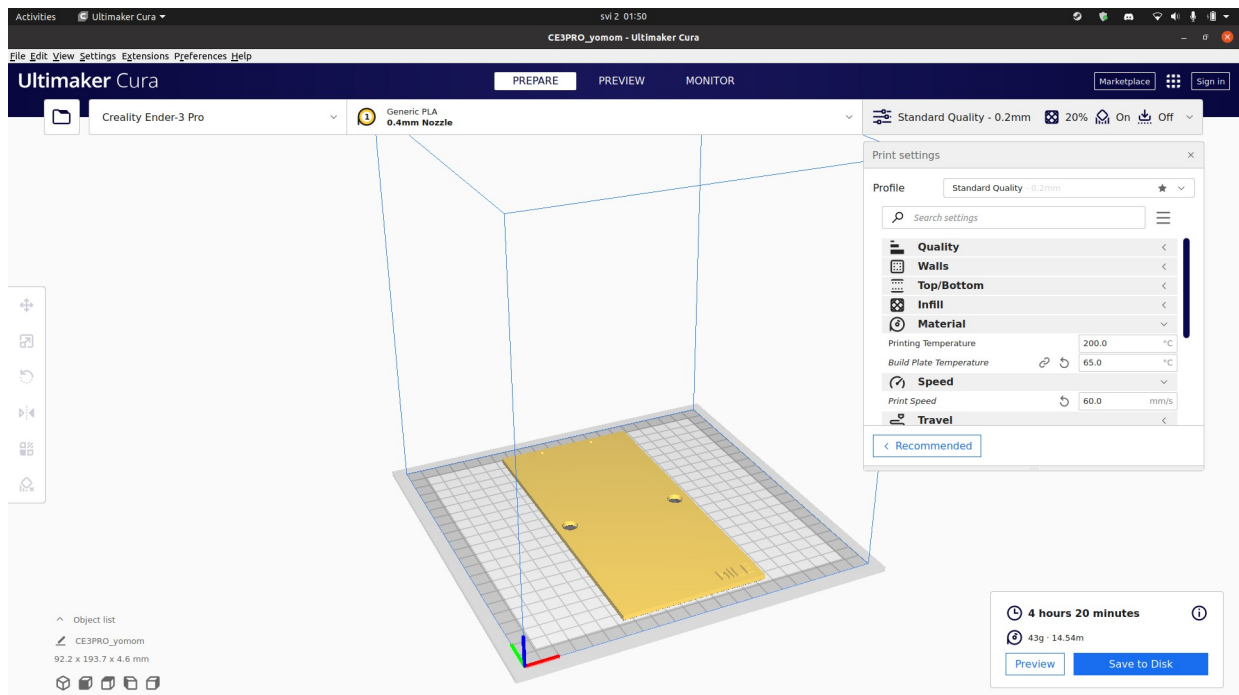
Creality Ender 3 Pro je 3D printer koji svojim karakteristikama omogućuje jednostavno i lako 3D printanje modela. Ovaj 3D printer je DIY printer tj. dolazi u već sastavljenim modulima koje sam trebala sastaviti u konačan 3D printer, pritegnuti i podesiti kako bi imali uspješno sklopljen 3D printer, a time i uspješan print.



Slika 8. Ender 3 Pro web stranica

Uz pomoć Ender 3 Pro sam isprintala Miletovu šasiju. Trebalo je par pokušaja dok nisam namjestila podlogu printera na savršeno ravnu podlogu. Dokument sa modelom sam stavila u Ultimater Cura slicer i razrezala ga kako bih ga mogla isprintati.

Print je trebao trajati oko četiri i pol sata no trajao je tri sata i petnaest minuta. Potrošio je oko 15m ili 43g bijelog PLA filamneta. Podloga je bila zagrijana na 65°, a nozzl na 210°. Brzinu printa sam naknadno povećala sa 100 na 215.



*Slika 9. Ultimater Cura slicer*

## 5. PROGRAMIRANJE ROBOTA

### 5.1. PRETRAŽIVANJE DOKUMENTACIJE

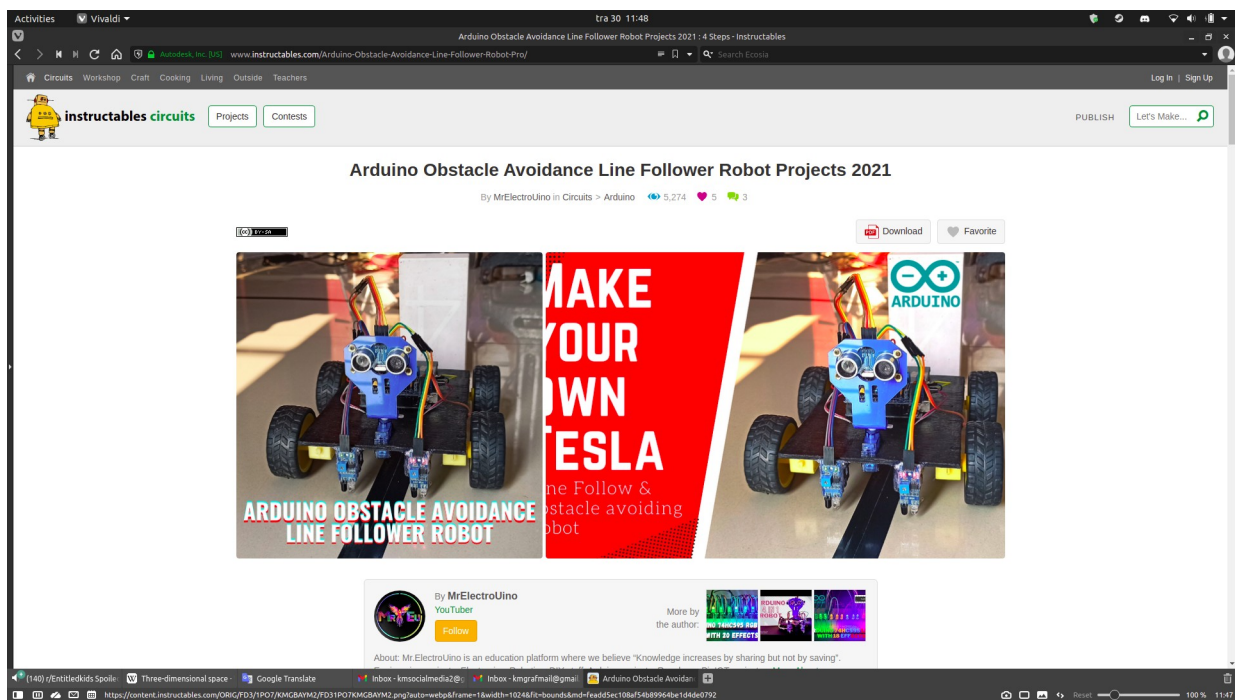
Ovo je bio prvi put da sam se susrela sa robotom koji autonomno nešto radi bez ikakvog vanjskog utjecaja osim linije. Pregledala sam službenu Arduino dokumentaciju kako bih dobila neki smisao koda. Arduino dokumentacija nema puno i nisu baš opširne. Samo za jednostavne projekte bi nam ove dokumentacije zapravo mogle pomoći.

Pretražila sam skoro svaki dio Git Hub i Stack Overflow-a bez imalo pomoći ili bar naslutnje kako započeti kod projekta.

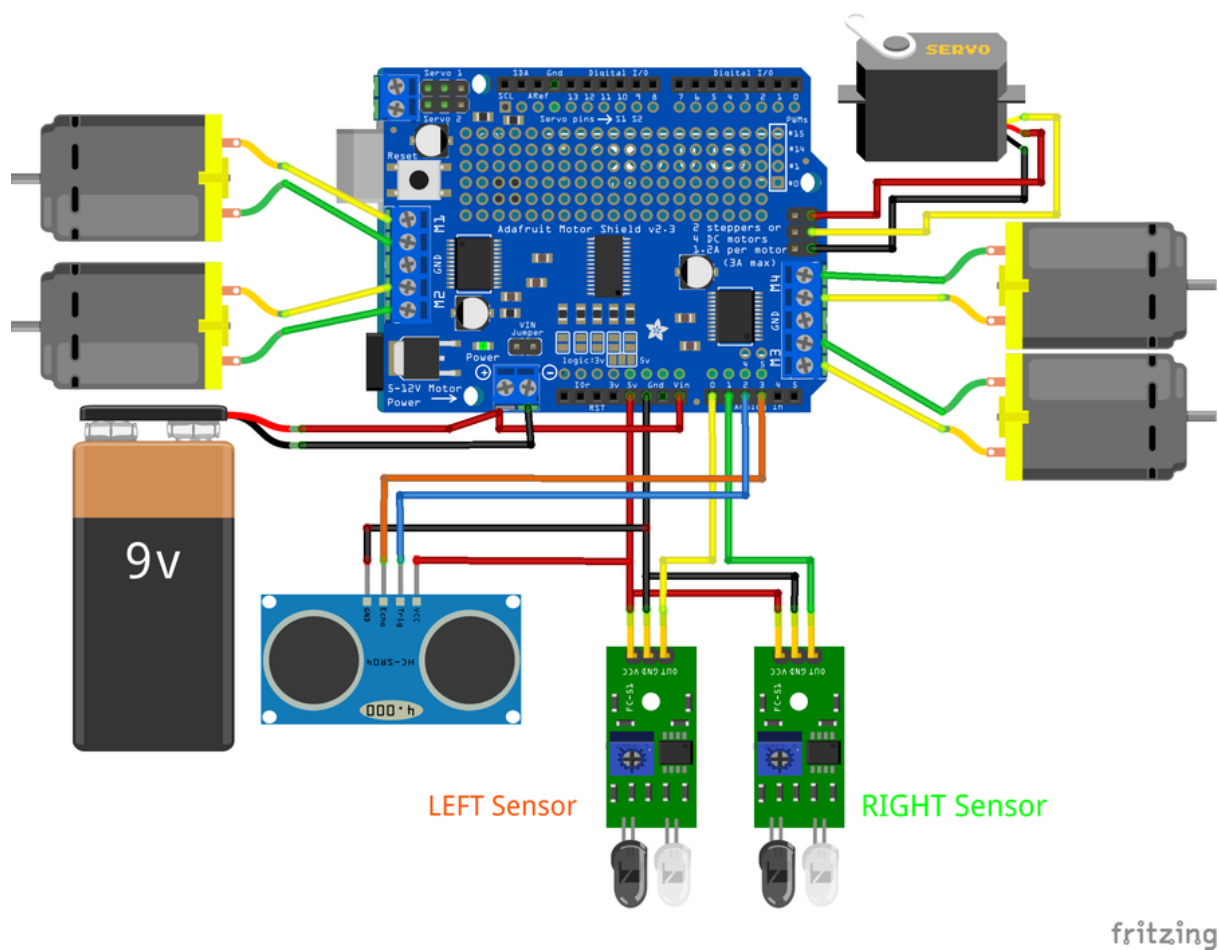
### 5.2. INSTRUCTABLES CIRCUITS

Pronašla sam Instructables circuits - „open source” stranice gdje su objavljeni projekti, 3D modeli te razni DIY projekti. Ljudi mogu postaviti, uređivati, skidati tuđe i svoje projekte kako bi pomogli zajednici.

Pronašla sam sličan projekt svome i modificirala sam njegovu šemu no ne i kod. Šemi sam morala maknuti ultrazvučni senzor jer nije bio dio projekta. Kod dobiven sa šemom nije radio te je bio prilagođen šemi sa ultrazvučnim senzorom. Pokušaj prilagodbe koda mojoj šemi nije uspio.



Slika 10. Instructables circuits web stranica



Slika 11. Primjer šema

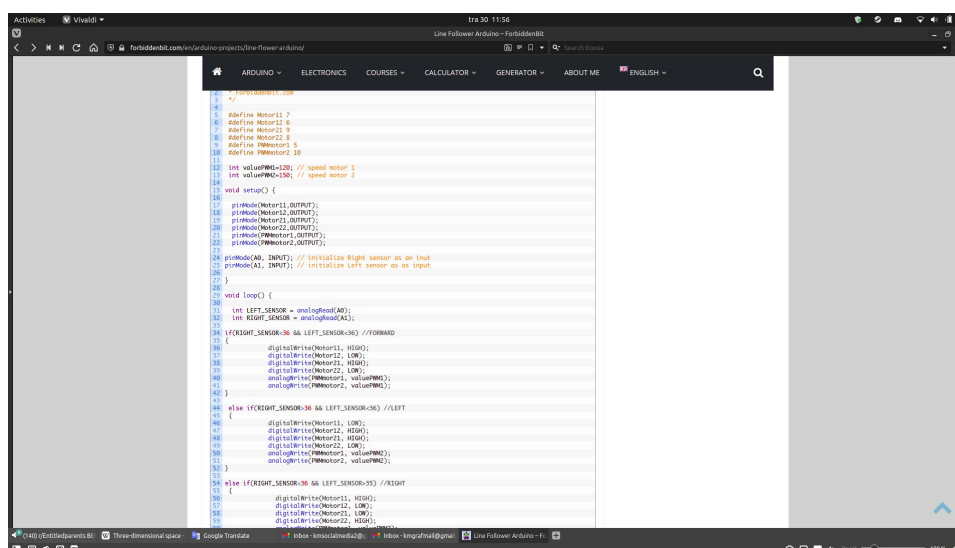
### 5.3. FORBIDDENBIT.COM

Forbiddenbit je također „open source“ stranica namijenjena za dijeljenje projekata. Posjetitelji stranice mogu dijeliti, pomagati drugima iz svijeta koji se žele asociirati sa „DIY“ zajednicom.

Upravo na Forbiddenbit stranici sam pronašla kod koji sam prilagodila svojim potrebama. Sadržavalo je više varijacija koda za različite mostove.



Slika 12. Forbiddenbit web stranica



Slika 13. Forbiddenbit kod

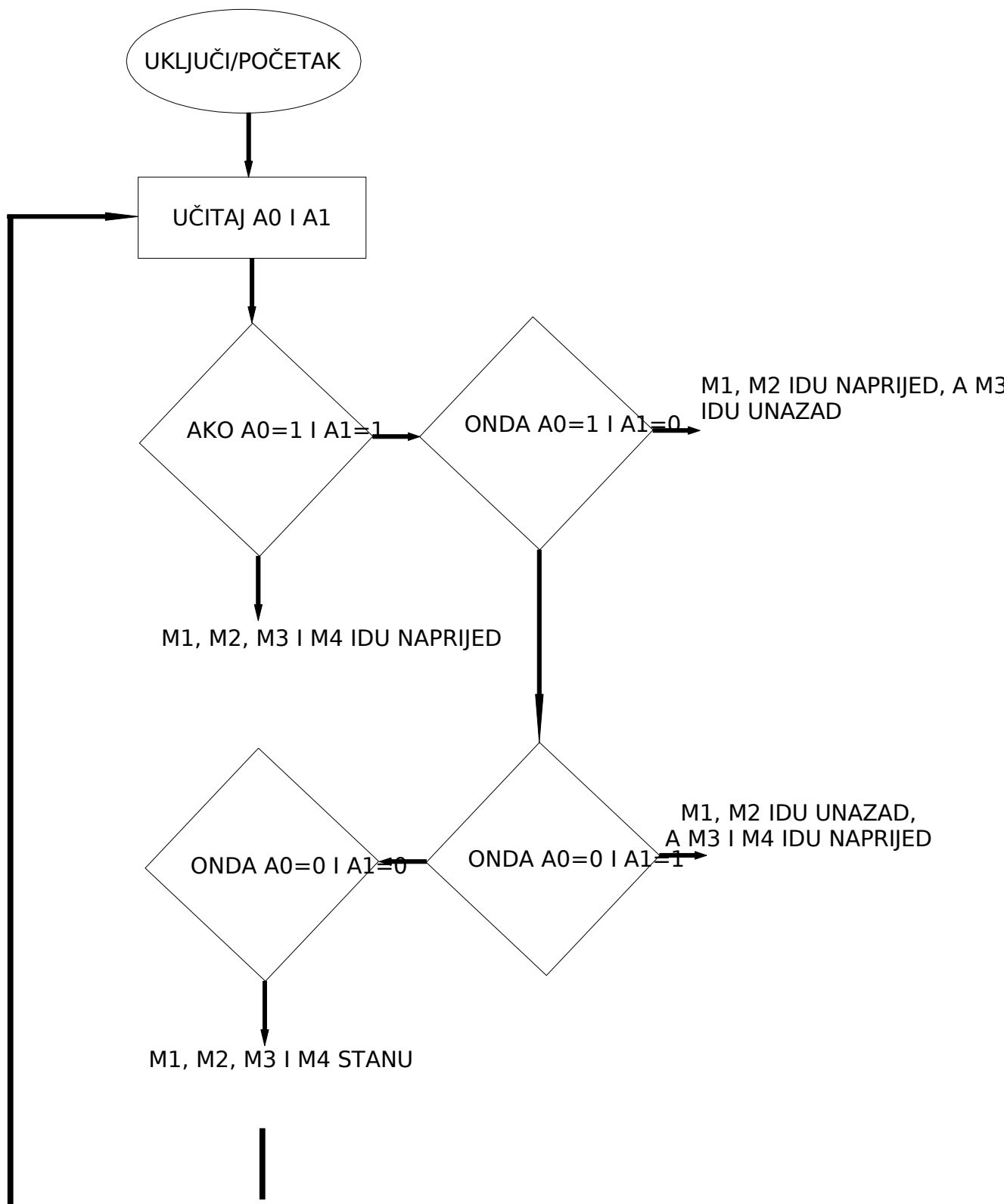
## 5.4. PRVI KOD

### 5.4.1. PSEDUDO JEZIK

Kod radi tako da kada pritisnete gumb za paljenje senzori počinju prepoznavati crnu i bijelu boju, to jest koliko se svjetlosti odbija s podloge. Ako se svjetlo upije (vidi crnu boju) to prepozna kao 0 i pošalje signal na Arduino Uno. Ako se svjetlo odbije (vidi bijelu boju) to prepozna kao 1 i pošalje signal na Arduino Uno. Nakon što Arduino Uno dobije informacije od oba senzora onda može započeti radnju.

Ako je lijevi senzor očitao 1 i desni senzor očitao 1 onda svi Servo motori idu prema naprijed. No, ako lijevi senzor očitao 1, a desni 0 onda dva lijeva kotača idu naprijed, a dva desna unazad što natjera Mileta da skreće u lijevu stranu. Također ako je lijevi senzor očitao 0 i desni senzor očitao 1 onda dva lijeva kotača idu unazad, a dva desna naprijed što natjera Mileta da skreće u desnu stranu. Kada bi Mile učitao 1 i 1 na oba senzora onda bi stao.

#### 5.4.2. DIJAGRAM TIJEKA





#### 5.4.3. PROGRAMSKI KOD

Kod koji sam pronašla na Forbiddenbit-u sa prenijela na Arduino Uno te testirala kako bih mogla vidjeti radi li. Njihov kod je izgledao ovako:

```
/*
 * Forbiddenbit.com
 */

#define Motor11 7
#define Motor12 6
#define Motor21 9
#define Motor22 8
#define PWMmotor1 5
#define PWMmotor2 10

int valuePWM1=120; // speed motor 1
int valuePWM2=150; // speed motor 2

void setup() {

    pinMode(Motor11,OUTPUT);
    pinMode(Motor12,OUTPUT);
    pinMode(Motor21,OUTPUT);
    pinMode(Motor22,OUTPUT);
    pinMode(PWMmotor1,OUTPUT);
    pinMode(PWMmotor2,OUTPUT);

    pinMode(A0, INPUT); // initialize Right sensor as an input
    pinMode(A1, INPUT); // initialize Left sensor as an input
```

```

}

void loop() {

    int LEFT_SENSOR = analogRead(A0);
    int RIGHT_SENSOR = analogRead(A1);

    if(RIGHT_SENSOR<36 && LEFT_SENSOR<36) //FORWARD
    {
        digitalWrite(Motor11, HIGH);
        digitalWrite(Motor12, LOW);
        digitalWrite(Motor21, HIGH);
        digitalWrite(Motor22, LOW);
        analogWrite(PWMmotor1, valuePWM1);
        analogWrite(PWMmotor2, valuePWM1);
    }

    else if(RIGHT_SENSOR>36 && LEFT_SENSOR<36) //LEFT
    {
        digitalWrite(Motor11, LOW);
        digitalWrite(Motor12, HIGH);
        digitalWrite(Motor21, HIGH);
        digitalWrite(Motor22, LOW);
        analogWrite(PWMmotor1, valuePWM2);
        analogWrite(PWMmotor2, valuePWM2);
    }

    else if(RIGHT_SENSOR<36 && LEFT_SENSOR>35) //RIGHT
    {
        digitalWrite(Motor11,
        HIGH);
        digitalWrite(Motor12, LOW);
        digitalWrite(Motor21, LOW);
        digitalWrite(Motor22, HIGH);
    }

```

```

        analogWrite(PWMmotor1,
        valuePWM2);
        analogWrite(PWMmotor2, valuePWM2);
    }

else if(RIGHT_SENSOR>35 && LEFT_SENSOR>35) //BACK
{
    digitalWrite(Motor11,
    LOW);
    digitalWrite(Motor12,
    LOW);
    digitalWrite(Motor21,
    LOW);
    digitalWrite(Motor22,
    LOW); delay(10000);
}
}

```

## 5.5. POPRAVAK KODA

Nakon analiziranja koda greške koje su se ponavljale su bili uskličnici ispred „RIGHT\_SENSOR” i „LEFT\_SENSOR” u varijablama koje su samo poništili varijablu te njihov zadatak. Također „DELAY” je usporavao motore previše što je uzrokovalo stajanje motora kod skretanja.

Zbog previše grešaka i nepotrebnih informacija kod nije funkcionirao za ovaj projekt. Odlučila sam uzeti kostur i ostatak koda napisati sama. Završna verzija koda kojeg i dalje aktivno koristim izgleda ovako:

```
//including the libraries
```

```
#include <AFMotor.h>
```

```
//defining pins and variables
```

```
#define left A0
```

```
#define right A1
```

```
//defining motors
```

```
AF_DCMotor motor1(1,  
MOTOR12_1KHZ);
```

```
AF_DCMotor motor2(2,  
MOTOR12_1KHZ);
```

```
AF_DCMotor motor3(3,  
MOTOR34_1KHZ);
```

```
AF_DCMotor motor4(4,  
MOTOR34_1KHZ);
```

```
void setup() {  
    //declaring pin types  
    pinMode(left,INPUT);  
    pinMode(right,INPUT);  
  
    //begin serial communication  
    Serial.begin(9600);  
  
}
```

```
void loop(){  
    //printing values of the  
    sensors to the serial monitor  
  
    Serial.println(digitalRead(left))  
    ;  
  
    Serial.println(digitalRead(right  
    ));
```

```
    //line detected by both  
    if(digitalRead(left)==1 &&  
    digitalRead(right)==1){  
        //Forward
```

```

    motor1.run(FORWARD);
    motor1.setSpeed(250);
    motor2.run(FORWARD);
    motor2.setSpeed(250);
    motor3.run(FORWARD);
    motor3.setSpeed(250);
    motor4.run(FORWARD);
    motor4.setSpeed(250);
}

//line detected by left sensor

else if(digitalRead(left)==1
&& digitalRead(right)==0){
    //turn left

    motor1.run(FORWARD);
    motor1.setSpeed(200);
    motor2.run(FORWARD);
    motor2.setSpeed(200);
    motor3.run(BACKWARD);
    motor3.setSpeed(200);
    motor4.run(BACKWARD);
    motor4.setSpeed(200);

}

//line detected by right
sensor

```

```

else if(digitalRead(left)==0
&& digitalRead(right)==1){

    //turn right

    motor1.run(BACKWARD);

    motor1.setSpeed(200);

    motor2.run(BACKWARD);

    motor2.setSpeed(200);

    motor3.run(FORWARD);

    motor3.setSpeed(200);

    motor4.run(FORWARD);

    motor4.setSpeed(200);

}

//line detected by none

else if(digitalRead(left)==0
&& !digitalRead(right)==0){

    //stop

    motor1.run(RELEASE);

    motor1.setSpeed(0);

    motor2.run(RELEASE);

    motor2.setSpeed(0);

    motor3.run(RELEASE);

    motor3.setSpeed(0);

    motor4.run(RELEASE);

    motor4.setSpeed(0);

```

}

}



## 5.6. TESTIRANJE

Nakon stavljanja šeme i prenošenja koda na Arduino Uno, Mile je radio, ali netočno. Dva od četiri kotača su išla u drugu stranu te su bili veoma spori. Žice su ispadale iz označenih mjesta te su se spajale sa željeznom podlogom što je uzrokovalo kratki spoj.

## 5.7. ZAVRŠNO USAVRŠAVANJE

Pregledavši cijeli sklop i kod shvatila sam da su neke od žica krivo okrenute, + i - su bili zamijenjeni. Brzinu sam u kodu povećala sa 100 na 200 te sam pričvrstila sve žice na mostu radi sigurnosti. Svaku žicu koja nije bila izolirana sam izolirala te sam sve stavila na 3D isprintanu podlogu koju sam izmodelirala.

Liniju sam napravila koristeći dva A3 papira koja sam zaljepila skupa bočno , te sam sa izolir trakom iscrtala liniju koju će Mile pratiti. Linija ima nekoliko skretanja te na kraju ima veliki crni blok koji označava kraj.

## 6. ZAKLJUČAK

Izrađujući „Mileta” stekla sam mnoga znanja i vještine koje će mi pomoći u daljnjem školovanju te također u životu. Samostalno sam naučila C++ programski jezik. Znanje robotike, 3D printanja i modeliranja koje sam stekla radeći robota je neprocijenivo. Svi izazovi s kojima sam se susrela su me naučili mnogo lekcija.

Najteži dio je upravo bio onaj najlakši, kotači. Nisu htjeli surađivati te je bilo previše trenja, a i neravna metalna šasija je stvarala dodatno opterećenje nad kotačima. Kako bih riješila taj problem naručila sam nove kotače i izmodelirala 3D šasiju. Što mi samo pokazuje da se svaka zapreka može prijeći.

Mile se može unaprijeđivati i može biti od puno više koristi nego što je sada kao jednostavan pratitelj linije. Robotika je teška i opširna te treba imati strpljena za nju. Rad na Miletu me naučio mnogo lekcija od kojih je najbitnije strpljenje i upornost.

## LITERATURA, IZVORI SA INTERNETA

- Massimo Banzi, Arduino web stranica
- <https://www.arduino.cc>
  
- Arduino web stranica, Dokumentacija
- <https://www.arduino.cc/en/main/docs>
  
- Tom Preston-Werner, Chris Wanstrath, Scott Chacon, P. J. Hyett, GitHub
- <https://github.com>
  
- Jarek, Forbiddenbit
- <https://forbiddenbit.com/en/arduino-projects/line-flower-arduino/>
  
- MrElectroUino, Instructables
- <https://www.instructables.com/Arduino-Obstacle-Avoidance-Line-Follower-Robot-Pro/>
  
- Line Follower Obstacle Avoidance Tracking IR Sensor for Robots
- <https://universal-solder.ca/product/line-follower-obstacle-avoidance-tracking-ir-sensor-robots-arduino-etc/>
  
- 3D program za modeliranje Tinkercad
- <https://www.tinkercad.com>
  
- Ender 3 Pro web stranica
- <https://www.creality.com/products/ender-3-pro-3d-printer>
  
- Ultimater Cura Slicer web stranica
- <https://ultimaker.com/software/ultimaker-cura>

## POPIS SLIKA

Slika 1. Arduino logo	2
Slika 2. Arduino mikrokontroler	2
Slika 3. C programski jezik logo	3
Slika 4. C++ programski jezik logo	3
Slika 5. Dijelovi Mileta	4
Slika 6. Tinkercad	6
Slika 7. Dizajn šasije u Tinkercadu	7
Slika 8. Ender 3 Pro web stranica	8
Slika 9. Ultimater Cura slicer	9
Slika 10. Instructables circuits web stranica	11
Slika 11. Primjer šeme	11
Slika 12. Forbiddenbit web stranica	12
Slika 13. Forbiddenbit kod	12