

Kosilica za košnju trave bazirana na Arduino Mega 2560

Vagan, Nenad

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:110:949148>

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

Download date / Datum preuzimanja: **2024-12-23**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVA

NENAD VAGAN

KOSILICA ZA KOŠNJU TRAVE BAZIRANA NA ARDUINO MEGA
2560

ZAVRŠNI RAD

ČAKOVEC, 2022.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVA

NENAD VAGAN

KOSILICA ZA KOŠNJU TRAVE BAZIRANA NA ARDUINO MEGA
2560

LAWN MOWER BASED ON ARDUINO MEGA 2560

ZAVRŠNI RAD

Mentor: Jurica Trstenjak,

dipl. ing.

ČAKOVEC, 2022.

ZAHVALA

Htio bih izraziti nekoliko riječi zahvale svima koji su mi omogućili da danas budem tu gdje jesam, te onima koji su mi pomagali prilikom izrade ovog rada. Svakako bih prvo želio zahvaliti svome mentoru Jurici Trstenjaku, v. pred., koji mi je pružio veliku čast omogućivši izradu ovog rada pod svojim vodstvom. Hvala Vam na posvećenom vremenu, znanju i profesionalnom usmjeravanju. Ujedno zahvaljujem i svim zaposlenicima Međimurskog veleučilišta u Čakovcu, koji su mi pomogli u stjecanju znanja.

Posebno zahvaljujem svojim roditeljima, bratu, zaručnici i prijateljima, koji su vjerovali u mene od samog početka. Od srca vam hvala na neizmjerne ljubavi i podršci koju ste mi pružali tijekom mog obrazovanja.

Nenad Vagan

SAŽETAK

U ovom završnom radu prikazan je i izrađen model kosilice baziran na Arduino Mega 2560, sa šesnaest ultrazvučnih senzora HC-SR04 za detekciju zapreka, bluetooth senzorom za upravljanje, OLED ekranom za prikaz temperature, vlage i indeksa topline. Pogon kosilice će biti na dva motora (12v 25rpm) spojena preko motor drivera Cytron MDD10A. Sama konstrukcija kosilice bit će na limu 600×400 mm² debljine 1 mm. Oklop će biti napravljen u stilu "Cyber Trucka". Oklop će biti izrađen od šperploče debljine 6 mm. Na oklop je stavljena solarna ploča, koja napaja bateriju, a je povezana na Arduino Mega2560 kontroler. Cijeli sustav je stavljen u zajedničko kućište. Cilj ovog projekta je prikazati tehnologiju daljinskog upravljanja robotske kosilice putem tehnologije koja omogućuje visok stupanj fleksibilnosti i personalizacije. Arduino nam omogućuje komunikaciju kosilice s pametnim uređajem ili računalom, korištenjem bluetootha. Čitav sklop je objedinjen u cjelinu. Paljenje sustava krećemo s podizanjem sklopke koja pokreće Arduino Mega2560 i sve ostale povezane dijelove. Nakon što se sustav podigne, slijedi povezivanje preko bluetooth-a na pametni uređaj ili računalo, te se pokreće aplikacija PathFinder. Odabire se željeni način rada - autonomni ili manualni. Prilikom pokretanja sustava, uključi se OLED ekran. Na početnom zaslonu piše je Međimursko veleučilište u Čakovcu te ime i prezime studenta. Nakon 5 sekundi, uključuje se sljedeći displej na kojem se prikazuje mjerenje temperature, vlage i indeksa topline sa senzora DHT11. Nakon povezivanja na bluetooth uključuju se druga sklopka, koja pokreće modul MDD10A, te motor za košnju trave 12v 3000RPM. U autonomnom načinu kosilica koristi 16 ultrazvučnih senzora za detekciju prepreka. Kada kosilica naiđe na prepreku, staje ili odabire najbolju putanju kojom bi zaobišla objekt ispred ili iza sebe. U manualnom načinu vožnje, operater upravlja pomoću raspoloživih tipki u aplikaciji, koje se prethodno podešavaju u kôdu. Kada kosilica krene prema objektu, automatski će se zaustaviti ispred ili iza objekta, na udaljenosti od 15cm.

Ključne riječi: *Arduino, Arduino Mega2560, OLED ekran, bluetooth, Motor Driver Cytron MDD10A, ultrazvučni senzor HC-SR04, DHT11 senzor vlage i temperature*

SADRŽAJ

1. Uvod	1
2. Cilj rada	1
3. Arduino platforma	2
4. Arduino Mega 2560	4
4.1 Karakteristike Arduino Mega 2560	5
5. Materijali potrebni za izradu završnog rada.....	6
5.1 Ultrazvučni senzor – HC-SR04.....	7
5.1.1. Izvorni kôd za ultrazvučni senzor – HC-SR04	8
5.2. Sklop za kontrolu dva motora Cytron MDD10A.....	12
5.2.1. Izvorni kôd za kontrolu dva motora Cytron MDD10A	13
5.3 <i>Bluetooth</i> modul HC-05	14
5.3.1. Izvorni kôd za <i>bluetooth</i> modul HC-05	15
5.4 OLED Joy-it SBC-OLED01	16
5.4.1. Izvorni kôd za OLED Joy-it SBC-OLED01	16
5.5 DC motor 12v 25 RPM	17
5.6 DC motor 12v 3000 RPM	18
5.7 Solarni panel 30W	19
5.8 Senzor temperature i vlage DHT11	20
5.8.1. Izvorni kôd za senzor temperature i vlage DHT11	21
6. Praktični rad	22
6.1 Izrada konstrukcije kosilice.....	23
6.1.1. Izrada poklopca kosilice.....	24
6.2. Spajanje Arduino Mega i ultrazvučnih senzora- HC-SR04.....	26
6.3. Spajanje Arduino Mega i sklopa za kontrolu dva motora MDD10A ..	28
6.4. Spajanje <i>bluetooth</i> modula HC-05 na Arduino Mega.....	29
6.5. Spajanje motora kosilice	30
6.5. Spajanje u cjelinu	31
7. Aplikacija za upravljanje kosilice	32

8. Zaključak	33
9. Popis literature.....	34
9.1. Popis slika	34
9.2. Popis tablica	36
10. Prilozi	37
10.1. Izvorni kôd za kosilicu	37

1. Uvod

Izrada i opis kosilice za košnju trave bazirane na Arduino Mega 2560, tema je ovog završnog rada. Radi se o kosilici sa šesnaest senzora blizine za detekciju zapreka, *bluetooth* senzorom za upravljanje, OLED ekranom za prikaz temperature, vlage i indeksa topline. Pogon kosilice je na dva motora (12v 25rpm) spojena preko *H-bridge* motor drivera Cytron MDD10A. Sama konstrukcija kosilice bit će na limu 600×400 mm² debljine 1 mm. Oklop će biti napravljen u stilu "*Cyber Trucka*". Cijeli sustav je stavljen u zajedničko kućište.

2. Cilj rada

Cilj ovog završnog rada je primijeniti stečeno znanje iz područja osnove elektrotehnike i elektronike, digitalnih elektroničkih sklopova i programiranja, u svrhu izrade kosilice za košnju trave bazirane na Arduino Mega 2560. Glavni zadatak je realizacija ovog projekta te upoznavanje čitatelja s mogućnostima i funkcijama Arduino platforme. Arduino platforma je odlično rješenje zbog svoje jednostavnosti i pristupačne cijene, što rezultira lakšim prikazom primjenjivanja spomenutog mehanizma.

3. Arduino platforma

Projekt Arduino nastao je 2003. godine u Italiji pod nazivom *Wiring*. Razvio ga je student Hernando Barragan u sklopu svog završnog rada. Cilj spomenutog projekta bio je napraviti jeftin i jednostavan alat za razvoj mnogobrojnih uređaja, koji vrše interakcije s okolinom pomoću raznih pokretača. Godine 2005. talijanska tvrtka Smart Projects je koristeći 8-bitne mikrokontrolere¹ Atmel AVR izradila je Arduino. Cilj im je bio stvoriti jednostavnu, malu i jeftinu platformu pomoću koje bi povezivanje računala s fizičkim svijetom bilo znatno lakše. Arduino se zbog svojih karakteristika koristi u nekoliko tisuća različitih projekata i mnogim drugim uređajima koje gotovo svakodnevno koristimo. Arduino softver je zbog svoje jednostavnosti idealan za početnike, a s druge strane dovoljno je prilagodljiv da zadovolji zahtjeve naprednih korisnika. Arduino softver radi na Windows², Mac³ i Linux⁴ platformama. Arduino se često koristi za izradu jeftinih znanstvenih instrumenata za različita područja ispitivanja, kao i za programiranje i robotiku.



Slika 1. Mikrokontroler Atmel AVR. Izvor: <https://humansthatmake.com/atmel-avr/>

(preuzeto 9.2.2021.)

Da bi Arduino obavio neku funkciju, mora biti programiran koristeći Arduino IDE, nakon čega se učitava u Arduino pločicu. Arduino IDE podržava programske jezike C i C++. Sučelje Arduino IDE je prikazano na slici (Slika 2.), gdje se vide dvije glavne

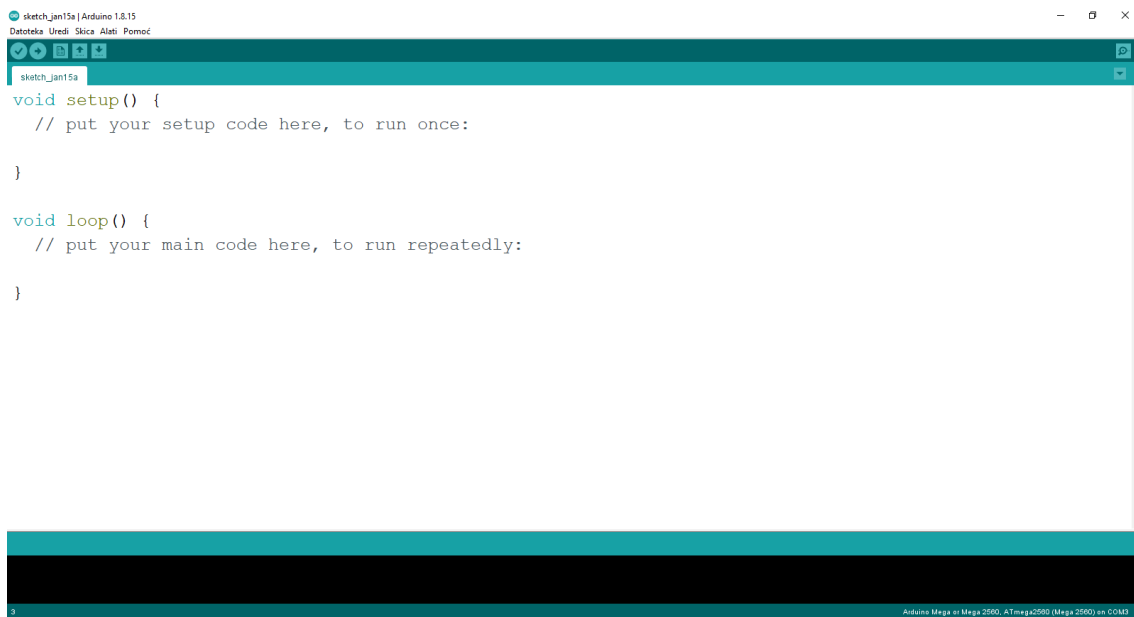
¹ Mikrokontroler- malo računalo na jednom čipu integriranog kruga metal-oksid-poluvodič.

² Windows- operacijski sustav tvrtke Microsoft

³ Mac- računala koje je razvila tvrtka Apple Computer

⁴ Linux- besplatni, open source operacijski sustav

funkcije koje moraju biti definirane u svakom Arduino programskom kôdu, a to su *setup()* i *loop()*. Funkcija *setup()* izvršava se jednom, kod pokretanja Arduina, potom slijedi funkcija *loop()* koja se neprestano poziva kada je Arduino uključen. Komunikacija s Arduinoom obavlja se pomoću USB-a⁵.

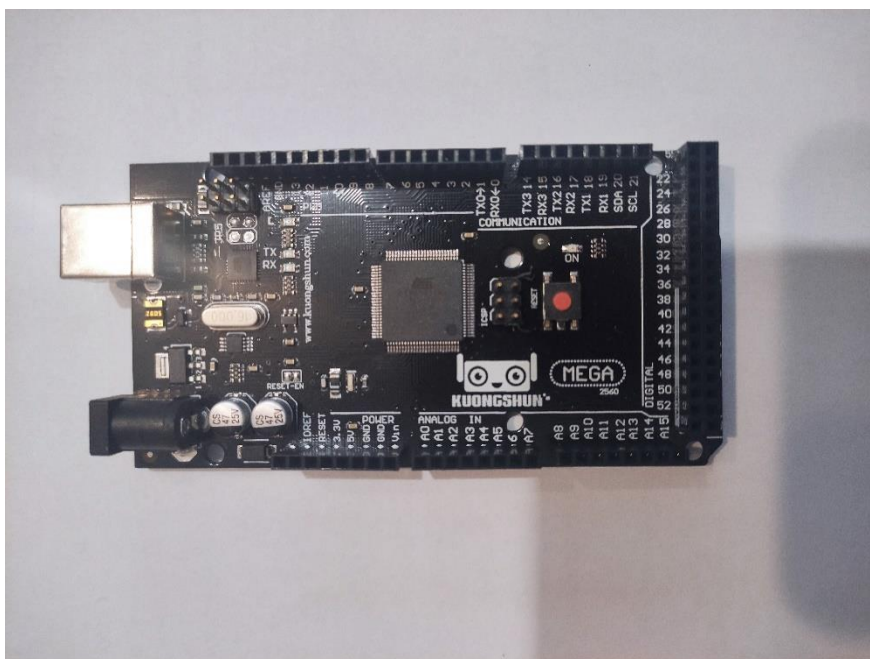


Slika 2. Korisničko sučelje Arduino IDE. Izvor: Nenad Vagan.

⁵ USB- Univerzalna serijska sabirnica

4. Arduino Mega 2560

Arduino Mega 2560 jedna je od najpoznatijih i najčešće korištenih Arduino ploča, jer predstavlja idealan izbor za početnike, ali i za zahtjevnije korisnike. Temeljen je na ATmega2560, a ima 54 digitalna ulazno/izlazna pina, od kojih se 15 može koristiti kao PWM⁶ izlaze, 16 analognih ulaza, 4 UART⁷-a, gumb za resetiranje, oscilator od 16 MHz⁸, USB priključak, utičnicu za napajanje, ICSP⁹ zaglavlje. Ploča Arduino Mega 2560 kompatibilna je s većinom modularnih pločica (Tablica 1.).



Slika 3. Arduino Mega 2560. Izvor: Nenad Vagan.

⁶ PWM - Modulacija širine impulsa ili modulacija trajanja impulsa (eng. Pulse-width modulation)

⁷ UART - serijski hardverski priključci

⁸ MHz - znak decimalne jedinice frekvencije megaherc

⁹ ICSP - Upetljano serijsko programiranje (eng. In-circuit serial programming)

4.1 Karakteristike Arduino Mega 2560

Tehničke karakteristike Arduino Mega pločice navedene su u Tablici 1.

Tablica 1. Tehničke karakteristike i dijelovi

Mikrokontroler	ATmega2560
Radni napon mikrokontrolera	5 V
Preporučeni ulazni napon	7-12 V
Maksimalan ulazni napon	6-20 V
Digitalni ulazno/izlazni pinovi	54 (od kojih 15 služe za PWM izlaz)
Analogni ulazni pinovi	16
Istosmjerna struja za pinove	20 mA
Istosmjerna struja za 3.3V pin	50 mA
Flash memorija	256 KB od kojih 8kb koristi za <i>bootloader</i>
SRAM memorija	8 KB
EEPROM memorija	4 KB
Procesor	16 MHz

Izvor: Nenad Vagan.

5. Materijali potrebni za izradu završnog rada

Komponente potrebne za izradu ovog završnog rada prikazane su u Tablici 2. Sve navedene komponente potrebno je spojiti u cjelinu.

Tablica 2. Komponente potrebne za izradu

Naziv na engleskom	Naziv na hrvatskom
<i>Ultrasonic Sensor - HC-SR04</i>	Ultrazvučni senzor - HC-SR04
<i>Motor driver Cytron MDD10A (H-bridge)</i>	Sklop za kontrolu dva motora Cytron MDD10A
<i>Bluetooth HC-05 transmission module</i>	<i>Bluetooth</i> modul za upravljanje
<i>Breadboard</i>	Pločica za spajanje
<i>OLED screen</i>	<i>OLED</i> ekran
<i>DC motor 12v 25rpm</i>	DC motor 12v, 25 okretaja u minuti
<i>DC motor 12V 3000rpm (for cutting grass)</i>	DC motor 12, 3000 okretaja u minuti (za košnju trave)
<i>Battery</i>	Baterija
<i>DHT11</i>	DHT senzor temperature, vlage
<i>Wires</i>	Žice
<i>Wheels</i>	Kotači

Izvor: Nenad Vagan.

5.1 Ultrazvučni senzor – HC-SR04

Jedan od najbitnijih dijelova ovog rada jest ultrazvučni senzor, koji se koristi za očitavanje udaljenosti. Koristi sonar za određivanje udaljenosti od predmeta, poput onoga kojeg koriste šišmiši. Senzor omogućuje otkrivanje dometa s velikom preciznošću, te stabilnim i točnim očitanjima. Karakteristike ultrazvučnog senzora HC-SR04 navedene su u Tablici 3.

Tablica 3. Karakteristike ultrazvučnog senzora HC-SR04

Napon	5 V
Domet	2cm – 400cm
Preciznost	3mm
Kut mjerenja	15 stupnjeva
Struja	2mA

Izvor: Nenad Vagan.



Slika 4. Ultrazvučni senzor- HC-SR04. Izvor:
Nenad Vagan.

5.1.1. Izvorni kôd za ultrazvučni senzor – HC-SR04

```
#include <NewPing.h>

#define trigger_pin_1 22
#define echo_pin_1 23
#define max_udaljenost 30
#define trigger_pin_2 24
#define echo_pin_2 25
#define max_udaljenost 30
#define trigger_pin_3 26
#define echo_pin_3 27
#define max_udaljenost 30
#define trigger_pin_4 28
#define echo_pin_4 29
#define max_udaljenost 30
#define trigger_pin_5 30
#define echo_pin_5 31
#define max_udaljenost 30
#define trigger_pin_6 32
#define echo_pin_6 33
#define max_udaljenost 30
#define trigger_pin_7 34
#define echo_pin_7 35
#define max_udaljenost 30
#define trigger_pin_8 36
#define echo_pin_8 37
#define max_udaljenost 30
#define trigger_pin_9 38
#define echo_pin_9 39
#define max_udaljenost 30
#define trigger_pin_10 40
#define echo_pin_10 41
#define max_udaljenost 30
#define trigger_pin_11 42
#define echo_pin_11 43
#define max_udaljenost 30
#define trigger_pin_12 44
#define echo_pin_12 45
#define max_udaljenost 30
#define trigger_pin_13 46
#define echo_pin_13 47
#define max_udaljenost 30
#define trigger_pin_14 48
#define echo_pin_14 49
#define max_udaljenost 30
#define trigger_pin_15 50
#define echo_pin_15 51
#define max_udaljenost 30
#define trigger_pin_16 52
#define echo_pin_16 53
#define max_udaljenost 30
```

```
NewPing sonar1(22, 23, 30); //prednji
NewPing sonar2(24, 25, 30); //prednji lijevi
NewPing sonar3(26, 27, 30); //prednji desni
NewPing sonar4(28, 29, 30); //lijevi
NewPing sonar5(30, 31, 30); //desni
NewPing sonar6(32, 33, 30); //lijevi dodatni
NewPing sonar7(34, 35, 30); //desni dodatni
NewPing sonar8(36, 37, 5); //lijevi bočni - prednji
NewPing sonar9(38, 39, 5); //desni bočni - prednji
NewPing sonar10(40, 41, 5); //lijevi bočni
NewPing sonar11(42, 43, 5); //desni bočni
NewPing sonar12(44, 45, 30); //stražnji lijevi
NewPing sonar13(46, 47, 30); //stražnji desni
NewPing sonar14(48, 49, 30); //stražnji dodatni sredina
NewPing sonar15(50, 51, 30); //stražnji lijevi dodatni
NewPing sonar16(52, 53, 30); //stražnji desni dodatni

if (Serial.available() > 0)
    serialData = Serial.read();
    if (serialData == 'A')
        goto Autonomno;

    if (serialData == 'F' && sonar1.ping_cm() != 0) {
        stani();
    }
    else if (serialData == 'F' && sonar1.ping_cm() != 0 ||
serialData == 'F' && sonar2.ping_cm() != 0) {
        naprijed_desno();
    }
    else if (serialData == 'F' && sonar1.ping_cm() != 0 ||
serialData == 'F' && sonar3.ping_cm() != 0) {
        naprijed_lijevo();
    }
    else if (serialData == 'F' && sonar2.ping_cm() != 0 ||
serialData == 'F' && sonar4.ping_cm() != 0 || serialData ==
'F' && sonar6.ping_cm() != 0 || serialData == 'F' &&
sonar8.ping_cm() != 0) {
        naprijed_desno();
    }
    else if (serialData == 'F' && sonar3.ping_cm() != 0 ||
serialData == 'F' && sonar5.ping_cm() != 0 || serialData ==
'F' && sonar7.ping_cm() != 0 || serialData == 'F' &&
sonar9.ping_cm() != 0) {
        naprijed_lijevo();
    }
    else if (serialData == 'F' && sonar10.ping_cm() != 0) {
        desno();
    }
    else if (serialData == 'F' && sonar11.ping_cm() != 0) {
        lijevo();
    }
    else if (serialData == 'F' && sonar8.ping_cm() != 0 ||
serialData == 'F' && sonar10.ping_cm() != 0) {
```



```
        naprijed();
    }
    else if (serialData == 'F' && sonar9.ping_cm() != 0 ||
serialData == 'F' && sonar11.ping_cm() != 0) {
        naprijed();
    }
    else if (serialData == 'F' && sonar1.ping_cm() == 0 ||
serialData == 'F' && sonar2.ping_cm() == 0 || serialData ==
'F' && sonar3.ping_cm() == 0 || serialData == 'F' &&
sonar4.ping_cm() == 0 || serialData == 'F' && sonar5.ping_cm()
== 0 || serialData == 'F' && sonar6.ping_cm() == 0 ||
serialData == 'F' && sonar7.ping_cm() == 0 || serialData ==
'F' && sonar8.ping_cm() == 0 || serialData == 'F' &&
sonar9.ping_cm() == 0) {
        naprijed();
    }
    else if (serialData == 'B' && sonar12.ping_cm() != 0 ||
serialData == 'B' && sonar13.ping_cm() != 0 || serialData ==
'B' && sonar14.ping_cm() != 0 || serialData == 'B' &&
sonar15.ping_cm() != 0 || serialData == 'B' &&
sonar16.ping_cm() != 0) {
        stani();
    }
    else if (serialData == 'B' && sonar10.ping_cm() != 0) {
        desno();
    }
    else if (serialData == 'B' && sonar11.ping_cm() != 0) {
        lijevo();
    }
    else if (serialData == 'B' && sonar10.ping_cm() != 0 ||
serialData == 'B' && sonar11.ping_cm() == 0 || serialData ==
'B' && sonar12.ping_cm() != 0 || serialData == 'B' &&
sonar13.ping_cm() == 0 || serialData == 'B' &&
sonar14.ping_cm() != 0 || serialData == 'B' &&
sonar15.ping_cm() == 0 || serialData == 'B' &&
sonar16.ping_cm() != 0) {
        nazad();
    }
    else if (serialData == 'B' && sonar10.ping_cm() != 0 ||
serialData == 'B' && sonar12.ping_cm() == 0 || serialData ==
'B' && sonar15.ping_cm() != 0) {
        nazad_desno();
    }
    else if (serialData == 'B' && sonar11.ping_cm() != 0 ||
serialData == 'B' && sonar13.ping_cm() != 0 || serialData ==
'B' && sonar16.ping_cm() != 0) {
        nazad_lijevo();
    }
    else if (serialData == 'B' && sonar9.ping_cm() != 0 ||
serialData == 'B' && sonar10.ping_cm() != 0) {
        nazad_desno();
    }
}
```

```
    else if (serialData == 'F' && sonar9.ping_cm() == 0 ||
serialData == 'F' && sonar10.ping_cm() == 0) {
        naprijed_lijevo();
    }
    else if (serialData == 'F' && sonar9.ping_cm() != 0 ||
serialData == 'F' && sonar11.ping_cm() != 0) {
        lijevo();
    }
    else if (serialData == 'F' && sonar8.ping_cm() == 0 ||
serialData == 'F' && sonar10.ping_cm() == 0) {
        desno();
    }
}
```

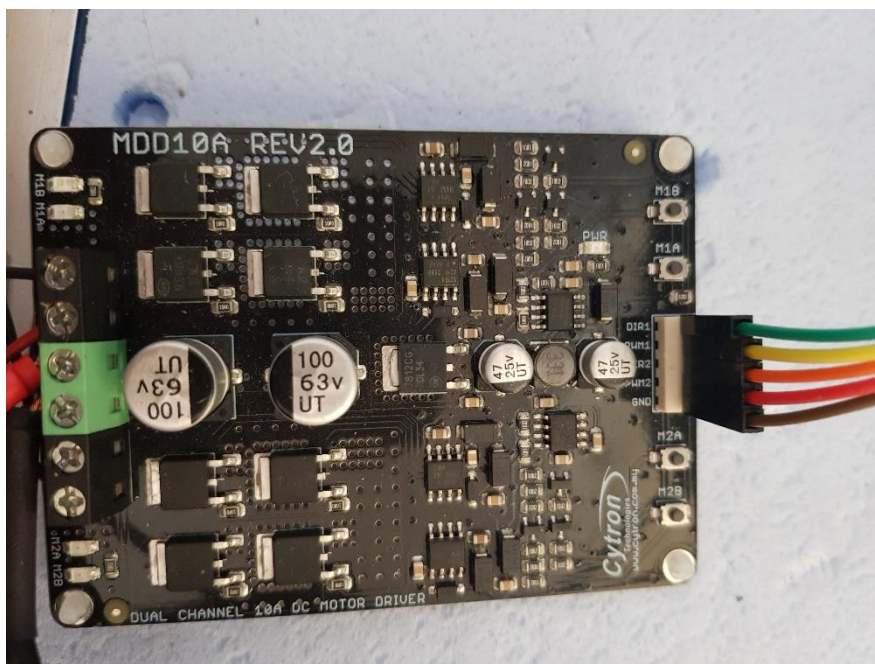
5.2. Sklop za kontrolu dva motora Cytron MDD10A

Sklop *MDD10A* omogućuje zamjenu polariteta na svojim izlazima, pa time postiže kontrolu okretanja motora u oba smjera na vrlo jednostavan način. Također, može upravljati i drugim uređajima, koji za određenu potrebu zahtijevaju zamjenu polariteta na svojim ulaznim krajevima.

Tablica 4. Karakteristike kontrolera motora MDD10A

Napon	5 – 30V (za motor), 5V za logiku
Dimenzije	84 mm × 62 mm × 23 mm
Struja	Max 10 A

Izvor: Nenad Vagan.



Slika 5. Motor driver Cytron MDD10A. Izvor: Nenad Vagan.

5.2.1. Izvorni kôd za kontrolu dva motora Cytron MDD10A

```
#define motor_dir1 8
#define motor_pwm1 9
#define motor_dir2 10
#define motor_pwm2 11

void stani()
{
  digitalWrite(motor_dir1, LOW);
  analogWrite(motor_pwm1, 0);
  digitalWrite(motor_dir2, LOW);
  analogWrite(motor_pwm2, 0);
}

void lijevo()
{
  digitalWrite(motor_dir1, LOW);
  analogWrite(motor_pwm1, 150);
  digitalWrite(motor_dir2, HIGH);
  analogWrite(motor_pwm2, 150);
}

void naprijed_lijevo()
{
  digitalWrite(motor_dir1, HIGH);
  analogWrite(motor_pwm1, 150);
  digitalWrite(motor_dir2, HIGH);
  analogWrite(motor_pwm2, 50);
}

void naprijed_desno()
{
  digitalWrite(motor_dir1, HIGH);
  analogWrite(motor_pwm1, 50);
  digitalWrite(motor_dir2, HIGH);
  analogWrite(motor_pwm2, 150);
}

void desno()
{
  digitalWrite(motor_dir1, HIGH);
  analogWrite(motor_pwm1, 150);
  digitalWrite(motor_dir2, LOW);
  analogWrite(motor_pwm2, 150);
}

void nazad_desno()
{
```

```
digitalWrite(motor_dir1, LOW);
analogWrite(motor_pwm1, 50);
digitalWrite(motor_dir2, LOW);
analogWrite(motor_pwm2, 150);

}
void nazad_lijevo()
{
    digitalWrite(motor_dir1, LOW);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, LOW);
    analogWrite(motor_pwm2, 50);

}

void naprijed()
{
    digitalWrite(motor_dir1, HIGH);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, HIGH);
    analogWrite(motor_pwm2, 150);

}

void nazad()
{
    digitalWrite(motor_dir1, LOW);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, LOW);
    analogWrite(motor_pwm2, 150);

}
```

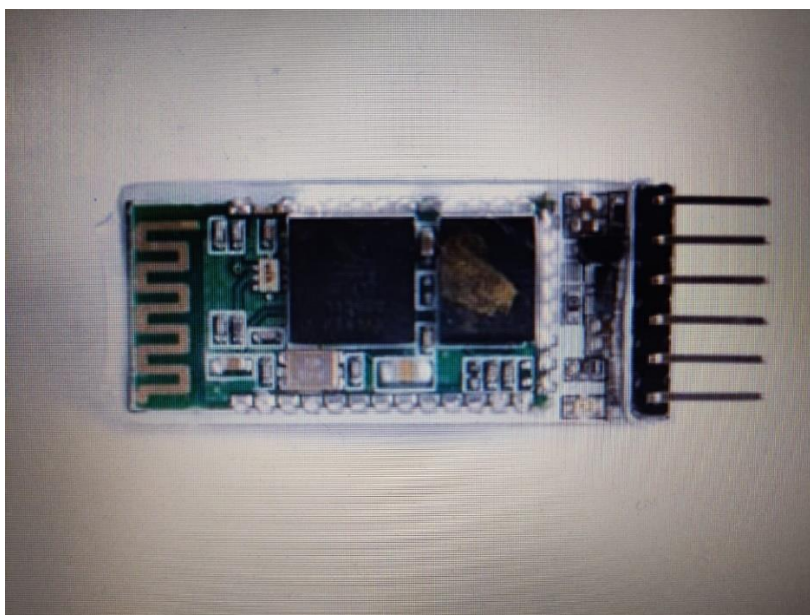
5.3 Bluetooth modul HC-05

Bluetooth modul HC-05 omogućuje serijsku komunikaciju između Arduina i drugog *bluetooth* uređaja (npr. računalo, laptop, Android mobitel) putem bežične *bluetooth* veze. S već izvučenim TX i RX linijama, spajanje s Arduinom je jednostavno. Moguće je uspostaviti i komunikaciju između dva Arduina putem *bluetooth* veze. Ovim modulom možete upravljati projektom putem Android uređaja ili primiti informacije s pločice na svoj pametni telefon.

Tablica 5. Karakteristike *bluetooth* modula HC-05

Napon	3.3 V
Dimenzije	37 mm × 17 mm
Način rada	Slave
Frekvencija	2.4GHz, Class 2
Specifikacija <i>bluetootha</i>	V2.0+EDR

Izvor: Nenad Vagan.

Slika 6. *Bluetooth* modul HC-05. Izvor: Nenad Vagan.

5.3.1. Izvorni kôd za *bluetooth* modul HC-05

```
void setup()
{
  Serial.begin(9600);

  if (Serial.available() > 0)
    serialData = Serial.read();
}
```

5.4 OLED Joy-it SBC-OLED01

OLED Joy-it SBC-OLED01 maleni je OLED ekran s crnom pozadinom i plavom bojom zaslona, koji se može kontrolirati pomoću I²C¹⁰, što otvara širok raspon aplikacija. Visoka razlučivost nudi oštru sliku za najviše zahtjeve s najmanjim potrebnim prostorom. Integrirani krug SSD1306 služi kao regulator, a zaslon radi s naponom od 5 V.



Slika 7. OLED ekran 0.96. Izvor: Nenad Vagan.

5.4.1. Izvorni kôd za OLED Joy-it SBC-OLED01

```
#include "U8glib.h"
u8g.firstPage();
do {
  u8g.setFont(u8g_font_helvB10);
  u8g.drawStr(20, 15, "Medimursko");
  u8g.drawStr(15, 30, "veleučilište u");
  u8g.drawStr(30, 45, "Čakovcu");
  u8g.drawStr(15, 60, "Nenad Vagan");
} while ( u8g.nextPage() );
delay(5000);
```

¹⁰ I²C – odnosi se na međusobno integrirani krug, odnosno inter-integrirani krug. Riječ je o serijskoj sabirnici za komunikaciju podataka.

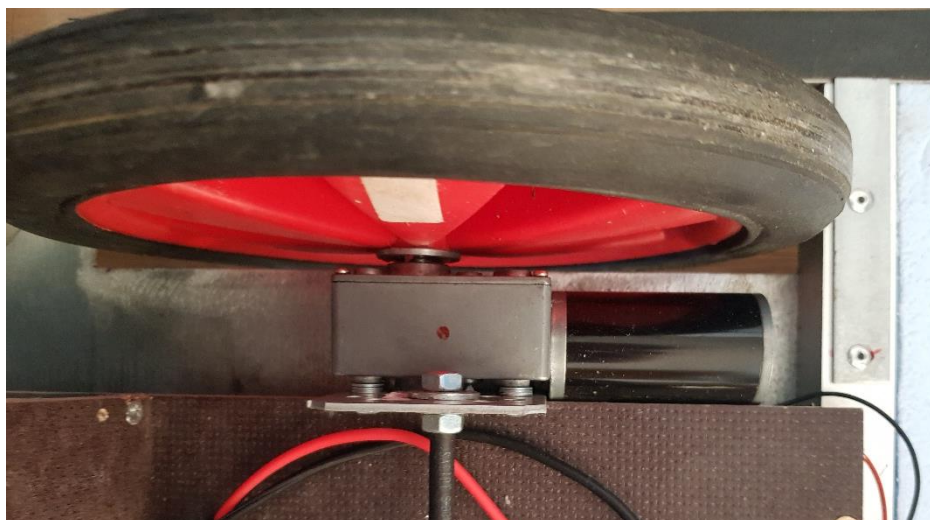
5.5 DC motor 12v 25 RPM

DC motor 12v 25 RPM koristi se za pokretanje pogona na kosilici, a njegove karakteristike prikazane u tablici 6.

Tablica 6. Karakteristike DC motora

Napon	6 – 12v
Snaga	3W
Broj okretaja	25 o/min
Dimenzije	80 mm × 25 mm × 21 mm

Izvor: Nenad Vagan.



Slika 8. DC motor 12v 25 o/min. Izvor: Nenad Vagan.

5.6 DC motor 12v 3000 RPM

DC motor 12v 3000 RPM istosmjerni je motor s 3000 o/min. Koristi za pokretanje diska na kosilici. Karakteristike motora prikazane su u tablici 7.

Tablica 7. Karakteristike DC motora 3000 o/min

Napon	12v
Snaga	25W
Broj okretaja	3000 o/min
Dimenzije	80 mm × 50 mm

Izvor: Nenad Vagan.



Slika 9. DC motor 12v 3000 o/min. Izvor: Nenad Vagan.

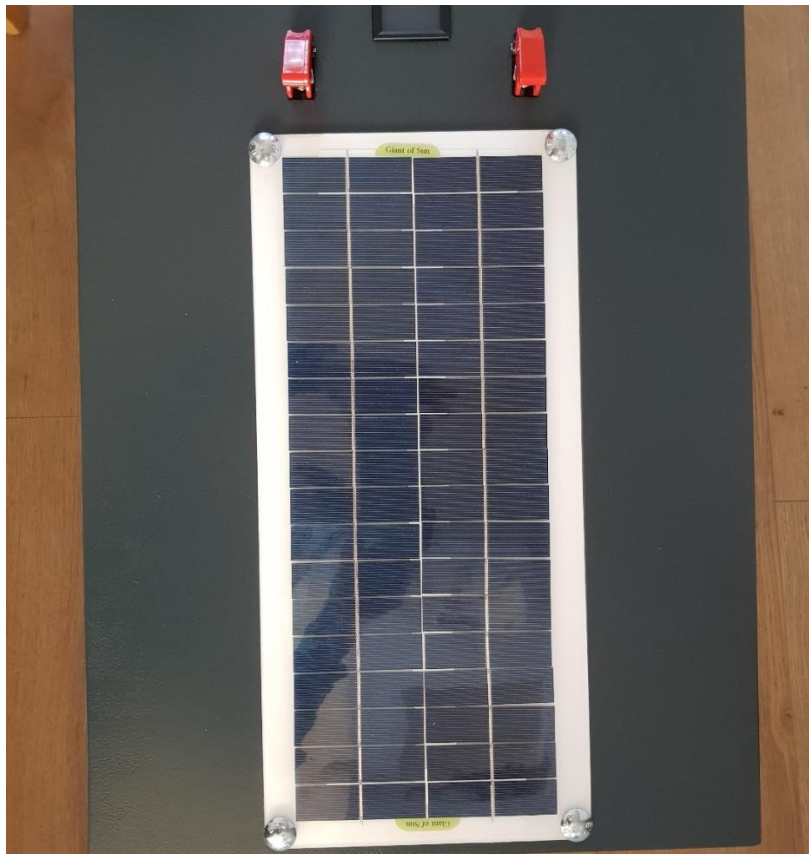
5.7 Solarni panel 30W

Solarni panel koristimo za dopunjavanje baterije koja potom napaja Arduino pločicu. Dimenzije solarne ploče su 435 mm × 200 mm × 2 mm, a težina je 100 g.

Tablica 8. Karakteristike solarnog panela

Napon	12v
Snaga	30W
Struja kratkog spoja	3A
Dimenzije	435 mm × 200 mm × 2 mm

Izvor: Nenad Vagan.



Slika 10. Solarni panel. Izvor: Nenad Vagan.

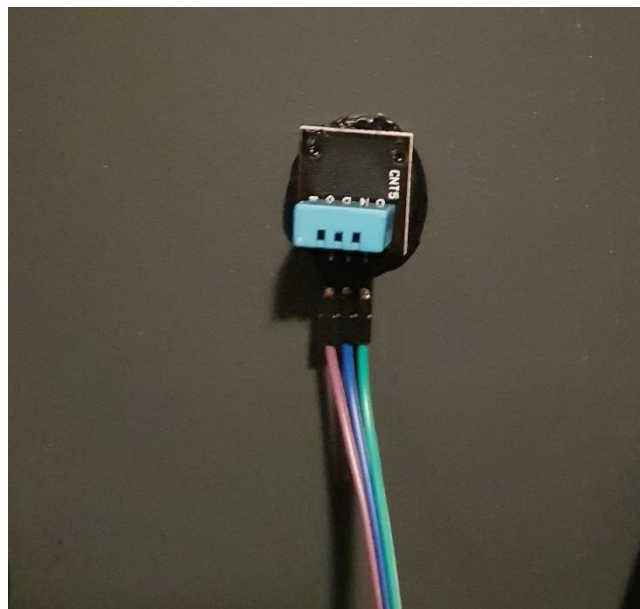
5.8 Senzor temperature i vlage DHT11

DHT11 je senzor temperature i vlage. Spaja se na Arduino vrlo jednostavno, putem samo jednog pina te vraća dvije varijable kao temperaturu i vlagu. Ne treba kalibraciju niti dodatno vrijeme da proradi.

Tablica 9. Karakteristike senzora DHT11

Ime pina	Opis
VCC	5V
GND	GND
SIGNAL	2

Izvor: Nenad Vagan.



Slika 11. DHT11 senzor temperature i vlage. Izvor: Nenad Vagan.

5.8.1. Izvorni kôd za senzor temperature i vlage DHT11

```
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
dht.begin();
u8g.firstPage();
do {
    u8g.setFont(u8g_font_helvR12);
    u8g.drawStr(0, 15, "Temp:");
    u8g.setPrintPos(75, 15);
    u8g.print(temp, 0);
    u8g.print((char)176);
    u8g.print("C");

    u8g.drawStr(0, 35, "Vlaga:");
    u8g.setPrintPos(75, 35);
    u8g.print(hum, 0);
    u8g.print("%");

    u8g.drawStr(0, 55, "Hi:");
    u8g.setPrintPos(75, 55);
    u8g.print(heat_indexC, 0);
    u8g.print("%");
} while ( u8g.nextPage() );
delay(5000);
```

6. Praktični rad

Praktični dio ovog završnog rada izrada je kosilice za košnju trave bazirane na Arduino Mega 2560. Način komuniciranja i spajanja pojedinih komponenata s Arduinom detaljno je opisan u ovom dijelu.



Slika 12. Kosilica u samom početku. Izvor: Nenad Vagan.



Slika 13. Konačan izgled kosilice. Izvor: Nenad Vagan.

6.1 Izrada konstrukcije kosilice

Konstrukcija kosilice izrađena je na limu debljine 1 mm, dimenzije 600×400 mm². Na izrezani lim stavljene su aluminijske cijevi za pojačanje same konstrukcije. Na konstrukciju su stavljena dva motora koja pokreću kosilicu i jedan kotač za okretanje. Izrada je započela s rezanjem aluminijskih cijevi pomoću kojih je dobiven željeni oblik. Nakon toga je prema obliku izrezan lim i pričvršćen na okvir. Motori su pričvršćeni na kutne profile. Držak akumulatora izrađen je od vodootporne i vatrootporne šperploče.



Slika 14. Izrada konstrukcije kosilice. Izvor: Nenad Vagan.



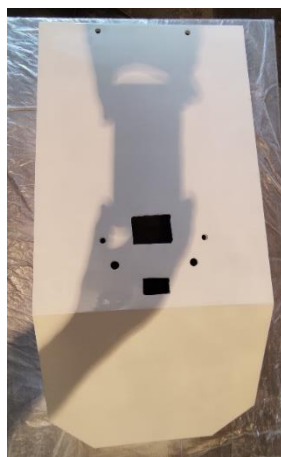
Slika 15. Gotova konstrukcija kosilice. Izvor: Nenad Vagan.

6.1.1. Izrada poklopca kosilice

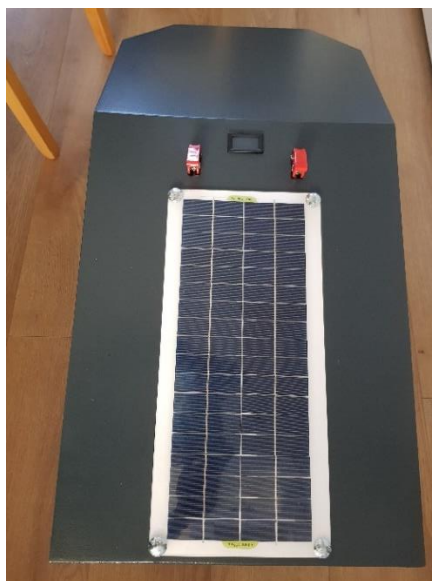
Poklopac kosilice izrađen je od vodootporne šperploče debljine 6 mm koja je kupljena po mjerama u specijaliziranoj trgovini. Šperploča je zalijepljena ljepilom za drvo i dodatno ojačana čavlicima. Nakon što se ljepilo osušilo, izbušene su rupe za senzore HC-SR04 i ostale dijelove. Dijelovi su prethodno izmjereni pomoću pomične mjerke kako bi razmaci između dijelova bili jednaki. Zatim se krenulo s kitanjem površine šperploče s ciljem dobivanja što ravnije površine. Nakon što se kit posušio krenulo se s brušenjem pomoću ekscentrične brusilice s brusnim papirom granulacije 120 pa sve do 240. Potom je slijedilo bojanje pomoću kompresorske šprice sa željenom bojom. Nakon što se završilo s izradom poklopca, krenulo se na stavljanje dijelova. Senzori HC-SR04, *bluetooth* i DHT11 fiksirani su pomoću vrućeg ljepila, a solarna ploča postavljena je direktno na poklopac i učvršćena pomoću četiri vijka. Prekidači su postavljeni u prethodno izbušene rupe i pričvršćeni dobivenim vijcima. Na kraju je poklopac pričvršćen na kućište kosilice pomoću okova za vrata.



Slika 16. Lijepljenje poklopca kosilice. Izvor: Nenad Vagan.



Slika 17. Priprema za bojanje kosilice. Izvor: Nenad Vagan.



Slika 18. Gotov poklopac kosilice. Izvor: Nenad Vagan.

6.2. Spajanje Arduino Mega i ultrazvučnih senzora- HC-SR04

Za ovaj rad koristio se 16 ultrazvučnih senzora HC-SR04. Na prednjoj strani nalazi se 7 senzora, dok su 4 senzora s bočne strane i 5 senzora na stražnjoj strani kosilice. Ultrazvučni senzor sastoji se od dva dijela :

- *Trigger* („okidač“) koji šalje zvučne signale
- *Echo* (refleksija) koja ih prima nazad

Dovođenjem napona na *Trigger*, pokreće se ultrazvučni transduktor¹¹ koji odašilje zvučne impulse, koji su iznad čujnosti ljudskog uha. Zvučni valovi putuju sve dok ne naiđu na prepreku, od koje se potom odbijaju. Mjeri se vrijeme refleksije koje registrira senzor, a podatak o vremenu se šalje preko *Echo* izvoda natrag u mikrokontroler na koji je spojen. Dobiveni signali se koriste kako bi se saznala udaljenost sljedećeg objekta. Taj signal je potrebno pretvoriti u podatke koji su nam prepoznatljivi. Dobiveni signal je vrijeme refleksije. U Tablici 10. su prikazani pinovi na Arduino pločici. Ultrazvučni senzor napaja se preko Arduino pločice s pina 5V.

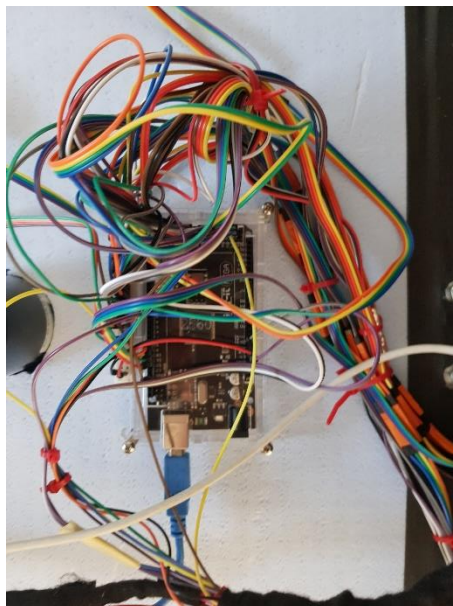
Tablica 10. Pozicija pinova na Arduino pločici

<i>Trigger</i> (okidač)	<i>Echo</i> (refleksija)
22	23
24	25
26	27
28	29
30	31
32	33
34	35
36	37
38	39
40	41
42	43
44	45
46	47

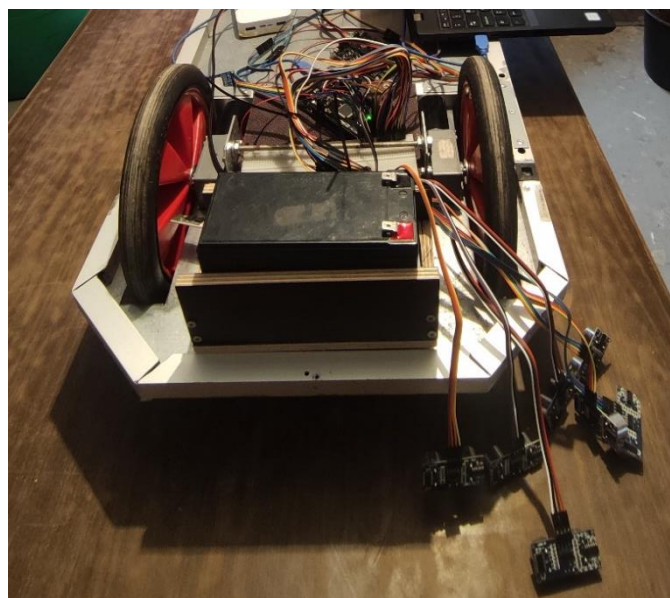
¹¹ Transduktor- generira izlaz na osnovu danog ulaza i/ili stanja koristeći akcije. Koristi se za upravljačke primjene.

48	49
50	51
52	53

Izvor: Nenad Vagan.



Slika 19. Spajanje senzora HC-SR04. Izvor: Nenad Vagan.



Slika 20. Spajanje senzora HC-SR04. Izvor: Nenad Vagan.

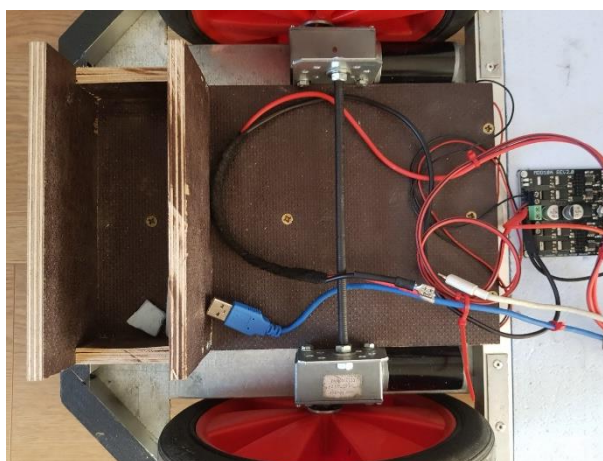
6.3. Spajanje Arduino Mega i sklopa za kontrolu dva motora MDD10A

MDD10A je dvostruki *H-Bridge* motorni kontroler koji omogućuje upravljanje brzinom i smjerom dvaju istosmjernih motora istovremeno. Kontroler može upravljati DC motorima koji imaju napone između 5 i 30V, sa strujom do 10A. Kontroler ima dva priključka za motore M1A, M1B, i M2A, M2B, napajanje za motor VCC, priključak za uzemljenje GND. Pinovi DIR1, DIR2, PWM1 i PWM2 služe za upravljanje motorima. PWM služi za kontrolu brzine vrtnje motora, a maksimalna brzina vrtnje je 255. DIR služi za usmjeravanje motora. U kôdu koristimo *LOW* i *HIGH*, u ovom slučaju *LOW* je nazad, *HIGH* je naprijed. Tablica 11. Pozicija MDD10A pinova na Arduino pločici.

Tablica 11. Pozicija MDD10A pinova na Arduino pločici

Ime pina	Opis
DIR1	8
PWM1	9
DIR2	10
PWM2	11
GND	GND

Izvor: Nenad Vagan.



Slika 21. Spajanje pogonskih motora na modul MDD10A. Izvor: Nenad Vagan.

6.4. Spajanje *bluetooth* modula HC-05 na Arduino Mega

Bluetooth modulom HC-05 može se komunicirati između dva mikrokontrolera poput Arduina ili s bilo kojim uređajem s *bluetooth* funkcijom, poput pametnog uređaja ili računala. U Google Play Storu postoje mnoge aplikacije putem kojih se spaja na *bluetooth* modul. Primjenom ovog modula mogu se prenijeti podatci s računala ili pametnih uređaja na mikrokontroler ili obrnuto. *Bluetooth* modul HC-05 ima dva načina rada; upravljački, kojim se upravlja ili mijenjaju postavke, te podatkovni, kojim se šalju ili primaju podatci s drugih *bluetooth* kompatibilnih uređaja. Prilikom prijenosa podataka s Arduino IDE *bluetooth* pinovi RX i TX moraju biti izbačene, jer u protivnom ne možemo prenijeti kôd. Tablica 12. Pozicija *bluetooth* pinova na Arduino pločici.

Tablica 12. Pozicija *bluetooth* pinova na Arduino pločici

Ime pina	Opis
VCC	5V
GND	GND
TX	RX
RX	TX

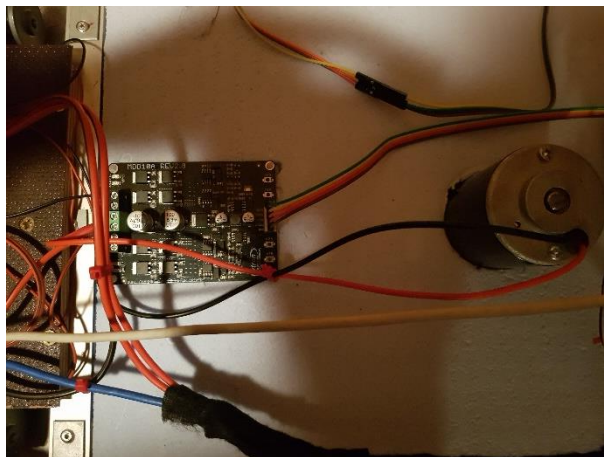
Izvor: Nenad Vagan.



Slika 22. Spajanje *bluetooth* senzora. Izvor: Nenad Vagan.

6.5. Spajanje motora kosilice

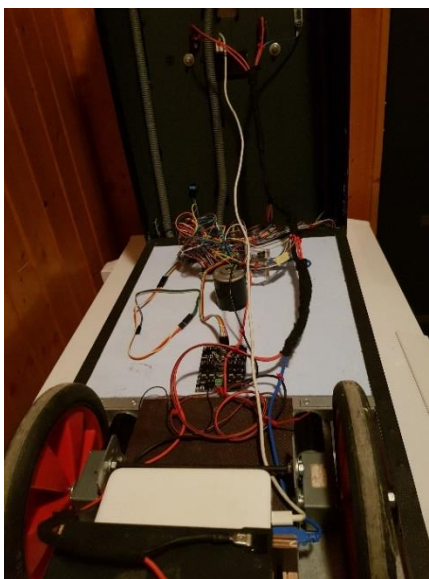
Motor kosilice spaja se u modul MDD10A na ulaze od akumulatora. Prilikom paljenja čitavog sklopa u isto se vrijeme uključuje i motor za košnju trave. Motor ima 3000 okretaja u minuti i samim time je pogodan za košnju trave. Na osovini motora pričvršćen je okrugli lim na kojem su pričvršćeni nožići od skalpela.



Slika 23. Spajanje motora kosilice. Izvor: Nenad Vagan.

6.5. Spajanje u cjelinu

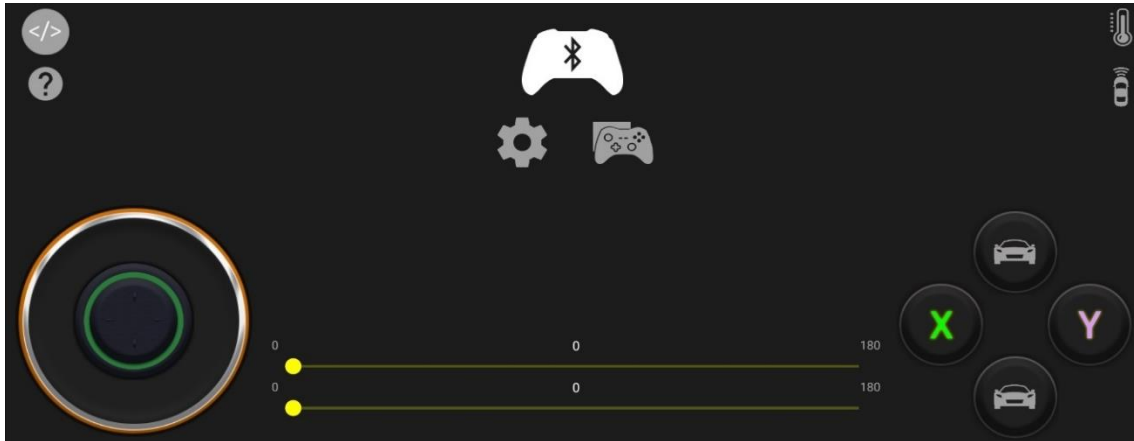
Za napajanje Arduina koristi se prijenosna baterija marke *XIAOMI* od 20000mAh koja se uključuje preko sklopke na kućištu. Prilikom paljenja upale se svi senzori osim MDD10A koji koristi jaču bateriju kako bi se mogli pokretati motori za pogon. Nakon što se spoji na *bluetooth*, na aplikaciji se odabire manualni način rada pa potom pali sklopku koja uključuje MDD10A. Sada je omogućeno upravljanje preko mobitela ili laptopa pomoću aplikacije *PathFinder* koja se može preuzeti s *Google Play* trgovine. Prilikom upravljanja u manualnom načinu, kosilica će se zaustaviti ili odabrati najbolju putanju kako ne bi udarila u objekt pred sobom ili iza sebe. U aplikaciji se može odabrati i autonomni način u kojem kosilica svojom putanjom upravlja sama pomoću 16 senzora. U kôdu je određena udaljenosti na koje senzori reagiraju, pa odabiru najbolju putanju. Na kućištu kosilice postavljen je i solarni panel koji napaja bateriju za Arduino.



Slika 24. Gotov sklop u cjelini. Izvor: Nenad Vagan.

7. Aplikacija za upravljanje kosilice

Za upravljanje kosilice koristi se aplikacija *PathFinder*, koja se može preuzeti s *Google Play* trgovine. Upravljanje se provodi pomoću tipki ili odabirom autonomne vožnje u kojoj se koristi ultrazvučni senzor za izbjegavanje prepreka.



Slika 25. Android aplikacija za upravljanje kosilicom. Izvor: Nenad Vagan.

8. Zaključak

Arduino platforma idealna je za izradu raznih projekata, zbog svoje pristupačne cijene i jednostavnosti izrade. U ovom završnom radu prikazana je izrada i realizacija robotske kosilice na Arduino platformi, putem koje se možemo povezati i upravljati. Arduino nam omogućuje čitanje stanja raznih senzora, za koje je potrebno napisati programski kôd, te ga učitati u mikrokontroler. Za upravljanje se koristi *PathFinder* mobilna aplikacija, a pametni telefon se povezuje s kosilicom pomoću Bluetooth modula HC-05. Za ovaj projekt postoji mogućnost nadogradnje s web kamerom, što bi omogućilo upravljanje kosilicom iz udobnosti doma. Također, moguće je dodati kompas i GPS sustav, kako bi kosilica mogla iscrtavati mapu po kojoj bi samostalno kosila bez prisustva čovjeka.

9. Popis literature

- [1] „Što je Arduino?“. E-elektro.blogspot.
<http://e-elektro.blogspot.com/2014/06/sto-je-arduino.html> (9.2.2021.)
- [2] Zenzerović P. (2014). *Arduino kroz jednostavne primjere*. 2. izd. Zagreb, Hrvatska zajednica tehničke kulture.
- [3] Bert van Dam (2016). *ARDUINO UNO-45 projekata za početnike i stručnjake*. Agencija Eho.
- [4] E-radionica.com. <https://e-radionica.com/hr/senzori.html> (15.4.2021.)
- [5] „Električni strojevi“. *Wikipedija*.
https://hr.wikipedia.org/wiki/Elektri%C4%8Dni_strojevi (11.5.2021.)

9.1. Popis slika

Slika 1. Mikrokontroler Atmel AVR. Izvor: Humans that make
<https://humansthatmake.com/atmel-avr/> (9.2.2021.)

Slika 2. Korisničko sučelje Arduino IDE. Izvor: Nenad Vagan.

Slika 3. Arduino Mega 2560. Izvor: Nenad Vagan.

Slika 4. Ultrazvučni senzor- HC-SR04. Izvor: Nenad Vagan.

Slika 5. Motor driver Cytron MDD10A. Izvor: Nenad Vagan.

Slika 6. *Bluetooth* modul HC-05. Izvor: Nenad Vagan.

Slika 7. OLED ekran 0.96. Izvor: Nenad Vagan.

Slika 8. DC motor 12v 25 o/min. Izvor: Nenad Vagan.

Slika 9. DC motor 12v 3000 o/min. Izvor: Nenad Vagan.

Slika 10. Solarni panel. Izvor: Nenad Vagan.

Slika 11. DHT11 senzor temperature i vlage. Izvor: Nenad Vagan.

Slika 12. Kosilica u samom početku. Izvor: Nenad Vagan.

Slika 13. Konačan izgled kosilice. Izvor: Nenad Vagan.

Slika 14. Izrada konstrukcije kosilice. Izvor: Nenad Vagan.

Slika 15. Gotova konstrukcija kosilice. Izvor: Nenad Vagan.

Slika 16. Lijepljenje poklopca kosilice. Izvor: Nenad Vagan.

Slika 17. Priprema za bojanje kosilice. Izvor: Nenad Vagan.

Slika 18. Gotov poklopac kosilice. Izvor: Nenad Vagan.

Slika 19. Spajanje senzora HC-SR04. Izvor: Nenad Vagan.

Slika 20. Spajanje senzora HC-SR04. Izvor: Nenad Vagan.

Slika 21. Spajanje pogonskih motora na modul MDD10A. Izvor: Nenad Vagan.

Slika 22. Spajanje *bluetooth* senzora. Izvor: Nenad Vagan.

Slika 23. Spajanje motora kosilice. Izvor: Nenad Vagan.

Slika 24. Gotov sklop u cjelini. Izvor: Nenad Vagan.

Slika 25. Android aplikacija za upravljanje kosilicom. Izvor: Nenad Vagan.

9.2. Popis tablica

Tablica 1. Tehničke karakteristike i dijelovi. Izvor: Nenad Vagan.

Tablica 2. Komponente potrebne za izradu. Izvor: Nenad Vagan.

Tablica 3. .Karakteristike ultrazvučnog senzora HC-SR04 Tablica. Izvor: Nenad Vagan.

Tablica 4. Karakteristike kontrolera motora MDD10A. Izvor: Nenad Vagan.

Tablica 5. Karakteristike *bluetooth* modula HC-05. Izvor: Nenad Vagan.

Tablica 6. Karakteristike DC motora. Izvor: Nenad Vagan.

Tablica 7. Karakteristike DC motora 3000 o/min. Izvor: Nenad Vagan.

Tablica 8. Karakteristike solarnog panela. Izvor: Nenad Vagan.

Tablica 9. Karakteristike senzora DHT11. Izvor: Nenad Vagan.

Tablica 10. Pozicija pinova na Arduino pločici. Izvor: Nenad Vagan.

Tablica 11. Pozicija MDD10A pinova na Arduino pločici. Izvor: Nenad Vagan.

Tablica 12. Pozicija *bluetooth* pinova na Arduino pločici. Izvor: Nenad Vagan.

10. Prilozi

10.1. Izvorni kôd za kosilicu

```
#include "U8glib.h"
#include <DHT.h>
#include <DHT_U.h>
#include <NewPing.h>
#include <Servo.h>
#ifdef __AVR__
#endif
Servo myservo;

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
U8GLIB_SH1106_128X64 u8g(U8G_I2C_OPT_NONE); // VDD=5V GND=GND
SCL=SCL SDA=SDA
#define PIN 4
#define SPEED 50
#define DELAYVAL 500
#define trigger_pin_1 22
#define echo_pin_1 23
#define max_udaljenost 30
#define trigger_pin_2 24
#define echo_pin_2 25
#define max_udaljenost 30
#define trigger_pin_3 26
#define echo_pin_3 27
#define max_udaljenost 30
#define trigger_pin_4 28
#define echo_pin_4 29
#define max_udaljenost 30
#define trigger_pin_5 30
#define echo_pin_5 31
#define max_udaljenost 30
#define trigger_pin_6 32
#define echo_pin_6 33
#define max_udaljenost 30
#define trigger_pin_7 34
#define echo_pin_7 35
#define max_udaljenost 30
#define trigger_pin_8 36
#define echo_pin_8 37
#define max_udaljenost 30
#define trigger_pin_9 38
#define echo_pin_9 39
#define max_udaljenost 30
#define trigger_pin_10 40
#define echo_pin_10 41
#define max_udaljenost 30
#define trigger_pin_11 42
```

```
#define echo_pin_11 43
#define max_udaljenost 30
#define trigger_pin_12 44
#define echo_pin_12 45
#define max_udaljenost 30
#define trigger_pin_13 46
#define echo_pin_13 47
#define max_udaljenost 30
#define trigger_pin_14 48
#define echo_pin_14 49
#define max_udaljenost 30
#define trigger_pin_15 50
#define echo_pin_15 51
#define max_udaljenost 30
#define trigger_pin_16 52
#define echo_pin_16 53
#define max_udaljenost 30

NewPing sonar1(22, 23, 30); //prednji
NewPing sonar2(24, 25, 30); //prednji lijevi
NewPing sonar3(26, 27, 30); //prednji desni
NewPing sonar4(28, 29, 30); //lijevi
NewPing sonar5(30, 31, 30); //desni
NewPing sonar6(32, 33, 30); //lijevi dodatni
NewPing sonar7(34, 35, 30); //desni dodatni
NewPing sonar8(36, 37, 5); //lijevi bočni - prednji
NewPing sonar9(38, 39, 5); //desni bočni - prednji
NewPing sonar10(40, 41, 5); //lijevi bočni
NewPing sonar11(42, 43, 5); //desni bočni
NewPing sonar12(44, 45, 30); //stražnji lijevi
NewPing sonar13(46, 47, 30); //stražnji desni
NewPing sonar14(48, 49, 30); //stražnji dodatni sredina
NewPing sonar15(50, 51, 30); //stražnji lijevi dodatni
NewPing sonar16(52, 53, 30); //stražnji desni dodatni

#define motor_dir1 8
#define motor_pwm1 9
#define motor_dir2 10
#define motor_pwm2 11

#define OB_daljina 50

int i = 0, pos = 0, trenutna_udaljenost = 0;
int daljina0 = 0, daljina30 = 0, daljina60 = 0, daljina85 = 0,
daljina90 = 0, daljina95 = 0, daljina120 = 0, daljina150 = 0,
daljina180 = 0 ;
unsigned long previous_millis = 0;
char serialData;
void setup()
```

```
{
  Serial.begin(9600);
  myservo.attach(4);
  dht.begin();
  u8g.firstPage();
  do {
    u8g.setFont(u8g_font_helvB10);
    u8g.drawStr(20, 15, "Medimursko");
    u8g.drawStr(15, 30, "veleučiliste u");
    u8g.drawStr(30, 45, "Čakovcu");
    u8g.drawStr(15, 60, "Nenad Vagan");
  } while ( u8g.nextPage() );
  delay(5000);

  pinMode(motor_dir1, OUTPUT);
  pinMode(motor_pwm1, OUTPUT);
  pinMode(motor_dir2, OUTPUT);
  pinMode(motor_pwm2, OUTPUT);
  analogWrite(motor_pwm1, 255);
  analogWrite(motor_pwm2, 255);
}

void stani()
{
  digitalWrite(motor_dir1, LOW);
  analogWrite(motor_pwm1, 0);
  digitalWrite(motor_dir2, LOW);
  analogWrite(motor_pwm2, 0);
}

void lijevo()
{
  digitalWrite(motor_dir1, LOW);
  analogWrite(motor_pwm1, 150);
  digitalWrite(motor_dir2, HIGH);
  analogWrite(motor_pwm2, 150);
}

void naprijed_lijevo()
{
  digitalWrite(motor_dir1, HIGH);
  analogWrite(motor_pwm1, 150);
  digitalWrite(motor_dir2, HIGH);
  analogWrite(motor_pwm2, 50);
}

void naprijed_desno()
{
```

```
digitalWrite(motor_dir1, HIGH);
analogWrite(motor_pwm1, 50);
digitalWrite(motor_dir2, HIGH);
analogWrite(motor_pwm2, 150);

}

void desno()
{
    digitalWrite(motor_dir1, HIGH);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, LOW);
    analogWrite(motor_pwm2, 150);
}

void nazad_desno()
{
    digitalWrite(motor_dir1, LOW);
    analogWrite(motor_pwm1, 50);
    digitalWrite(motor_dir2, LOW);
    analogWrite(motor_pwm2, 150);
}

void nazad_lijevo()
{
    digitalWrite(motor_dir1, LOW);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, LOW);
    analogWrite(motor_pwm2, 50);
}

void naprijed()
{
    digitalWrite(motor_dir1, HIGH);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, HIGH);
    analogWrite(motor_pwm2, 150);
}

void nazad()
{
    digitalWrite(motor_dir1, LOW);
    analogWrite(motor_pwm1, 150);
    digitalWrite(motor_dir2, LOW);
    analogWrite(motor_pwm2, 150);
}

int daljina(int pos)
{
    myservo.write(pos);
```

```
    delay(300);
    trenutna_udaljenost = sonar1.ping_cm() || sonar2.ping_cm()
    || sonar3.ping_cm() || sonar4.ping_cm() || sonar5.ping_cm() ||
    sonar6.ping_cm() || sonar7.ping_cm() || sonar8.ping_cm() ||
    sonar9.ping_cm() || sonar10.ping_cm() || sonar11.ping_cm() ||
    sonar12.ping_cm() || sonar13.ping_cm() || sonar14.ping_cm() ||
    sonar15.ping_cm() || sonar16.ping_cm();
    if (trenutna_udaljenost == 0)
        trenutna_udaljenost = 100;
    if (trenutna_udaljenost > 0 && trenutna_udaljenost < 30) {
        Serial.print("B");
        if (pos == 90) {
            nazad();
            delay(500);
        }
        if (pos < 90) {
            nazad_desno();
            delay(500);
        }
        if (pos > 90) {
            nazad_lijevo();
            delay(500);
        }
        return trenutna_udaljenost;
    }
}
void loop() {
    float hum = dht.readHumidity();
    float temp = dht.readTemperature();
    float fah = dht.readTemperature(true);
    if (isnan(hum) || isnan(temp) || isnan(fah)) {
        Serial.println("Neuspješno čitanje s DHT senzora");
        return;
    }
    float heat_index = dht.computeHeatIndex(fah, hum);
    float heat_indexC = dht.convertFtoC(heat_index);
    //Izračunavanje indeksa topline u Celzijusima

    u8g.firstPage();
    do {
        u8g.setFont(u8g_font_helvR12);
        u8g.drawStr(0, 15, "Temp:");
        u8g.setPrintPos(75, 15);
        u8g.print(temp, 0);
        u8g.print((char)176);
        u8g.print("C");

        u8g.drawStr(0, 35, "Vlaga:");
        u8g.setPrintPos(75, 35);
        u8g.print(hum, 0);
        u8g.print("%");
    } while (u8g.nextPage());
}
```



```
u8g.drawStr(0, 55, "Hi:");
u8g.setPrintPos(75, 55);
u8g.print(heat_indexC, 0);
u8g.print("%");
} while ( u8g.nextPage() );
delay(5000);

Autonomno: stani();
delay(300);

while (1) {
  if (Serial.available() > 0)
    serialData = Serial.read();
  if (serialData == 'M')
    goto Manualno;
above: daljina90 = 0;

  daljina90 = daljina(90);
  delay(10);

  while (daljina90 >= OB_daljina ) {

    if (millis() - previous_millis > 2000) {
      previous_millis = millis();
      daljina(180);
      delay(50);
      daljina(0);
      delay(50);
    }

    daljina90 = daljina(90);

    naprijed(); Serial.print("F");

    if (Serial.available() > 0)
      serialData = Serial.read();
    if (serialData == 'M')
      goto Manualno;
  }
  stani();

  if (daljina90 < OB_daljina)
  {
    stani();
    delay(100);
    daljina60 = 0;
    daljina60 = daljina(60);
    delay(200);
    daljina120 = 0;
    daljina120 = daljina(120);
    delay(200);
    if (Serial.available() > 0)
```

```
    serialData = Serial.read();
    if (serialData == 'M')
        goto Manualno;

    if (daljina60 > OB_daljina || daljina120 > OB_daljina )
    {
        if (daljina60 >= daljina120) {
            naprijed_desno(); Serial.print("R");
            delay(50); goto above;
        }
        else if (daljina60 < daljina120) {
            naprijed_lijevo(); Serial.print("L");
            delay(50); goto above;
        }
    }
    if (daljina60 < OB_daljina && daljina120 < OB_daljina)
    {
above1: daljina30 = 0;
        daljina30 = daljina(30);
        delay(200);
        daljina150 = 0;
        daljina150 = daljina(150);
        delay(200);
        if (Serial.available() > 0)
            serialData = Serial.read();
        if (serialData == 'M')
            goto Manualno;
        if (daljina30 > OB_daljina || daljina150 > OB_daljina
    )
        {
            if (daljina30 >= daljina150) {
                desno(); Serial.print("R");
                delay(100); goto above;
            }
            else if (daljina30 < daljina150) {
                lijevo(); Serial.print("L");
                delay(100); goto above;
            }
        }
    }
    if (daljina30 < OB_daljina && daljina150 < OB_daljina)
    {
        daljina0 = 0;
        daljina0 = daljina(0);
        delay(200);
        daljina180 = 0;
        daljina180 = daljina(180);
        delay(200);
        if (Serial.available() > 0)
            serialData = Serial.read();
        if (serialData == 'M')
            goto Manualno;
```

```
)
    if (daljina0 > OB_daljina || daljina180 > OB_daljina
)
    {
        if (daljina0 >= daljina180) {
            nazad_desno(); Serial.print("R");
            delay(200); goto above;
        }

        else if (daljina0 < daljina180) {
            nazad_lijevo(); Serial.print("L");
            delay(200); goto above;
        }
    }
    if (daljina0 < OB_daljina && daljina180 <
OB_daljina)
    {
        nazad(); Serial.print("B");
        delay(200);
        goto above1;
    }
}
}
Manualno: stani();
delay(300);
while (1) {
    if (Serial.available() > 0)
        serialData = Serial.read();
    if (serialData == 'A')
        goto Autonomno;
    if (serialData == 'F' && sonar1.ping_cm() != 0) {
        stani();
    }
    else if (serialData == 'F' && sonar1.ping_cm() != 0 ||
serialData == 'F' && sonar2.ping_cm() != 0) {
        naprijed_desno();
    }
    else if (serialData == 'F' && sonar1.ping_cm() != 0 ||
serialData == 'F' && sonar3.ping_cm() != 0) {
        naprijed_lijevo();
    }
    else if (serialData == 'F' && sonar2.ping_cm() != 0 ||
serialData == 'F' && sonar4.ping_cm() != 0 || serialData ==
'F' && sonar6.ping_cm() != 0 || serialData == 'F' &&
sonar8.ping_cm() != 0) {
        naprijed_desno();
    }
    else if (serialData == 'F' && sonar3.ping_cm() != 0 ||
serialData == 'F' && sonar5.ping_cm() != 0 || serialData ==
'F' && sonar7.ping_cm() != 0 || serialData == 'F' &&
sonar9.ping_cm() != 0) {
        naprijed_lijevo();
    }
}
```

```
    }
    else if (serialData == 'F' && sonar10.ping_cm() != 0) {
        desno();
    }
    else if (serialData == 'F' && sonar11.ping_cm() != 0) {
        lijevo();
    }
    else if (serialData == 'F' && sonar8.ping_cm() != 0 ||
serialData == 'F' && sonar10.ping_cm() != 0) {
        naprijed();
    }
    else if (serialData == 'F' && sonar9.ping_cm() != 0 ||
serialData == 'F' && sonar11.ping_cm() != 0) {
        naprijed();
    }
    else if (serialData == 'F' && sonar1.ping_cm() == 0 ||
serialData == 'F' && sonar2.ping_cm() == 0 || serialData ==
'F' && sonar3.ping_cm() == 0 || serialData == 'F' &&
sonar4.ping_cm() == 0 || serialData == 'F' && sonar5.ping_cm()
== 0 || serialData == 'F' && sonar6.ping_cm() == 0 ||
serialData == 'F' && sonar7.ping_cm() == 0 || serialData ==
'F' && sonar8.ping_cm() == 0 || serialData == 'F' &&
sonar9.ping_cm() == 0) {
        naprijed();
    }
    else if (serialData == 'B' && sonar12.ping_cm() != 0 ||
serialData == 'B' && sonar13.ping_cm() != 0 || serialData ==
'B' && sonar14.ping_cm() != 0 || serialData == 'B' &&
sonar15.ping_cm() != 0 || serialData == 'B' &&
sonar16.ping_cm() != 0) {
        stani();
    }
    else if (serialData == 'B' && sonar10.ping_cm() != 0) {
        desno();
    }
    else if (serialData == 'B' && sonar11.ping_cm() != 0) {
        lijevo();
    }
    else if (serialData == 'B' && sonar10.ping_cm() != 0 ||
serialData == 'B' && sonar11.ping_cm() == 0 || serialData ==
'B' && sonar12.ping_cm() != 0 || serialData == 'B' &&
sonar13.ping_cm() == 0 || serialData == 'B' &&
sonar14.ping_cm() != 0 || serialData == 'B' &&
sonar15.ping_cm() == 0 || serialData == 'B' &&
sonar16.ping_cm() != 0) {
        nazad();
    }
    else if (serialData == 'B' && sonar10.ping_cm() != 0 ||
serialData == 'B' && sonar12.ping_cm() == 0 || serialData ==
'B' && sonar15.ping_cm() != 0) {
        nazad_desno();
    }
}
```

```
        else if (serialData == 'B' && sonar11.ping_cm() != 0 ||
serialData == 'B' && sonar13.ping_cm() != 0 || serialData ==
'B' && sonar16.ping_cm() != 0) {
            nazad_lijevo();
        }
        else if (serialData == 'B' && sonar9.ping_cm() != 0 ||
serialData == 'B' && sonar10.ping_cm() != 0) {
            nazad_desno();
        }
        else if (serialData == 'F' && sonar9.ping_cm() == 0 ||
serialData == 'F' && sonar10.ping_cm() == 0) {
            naprijed_lijevo();
        }
        else if (serialData == 'F' && sonar9.ping_cm() != 0 ||
serialData == 'F' && sonar11.ping_cm() != 0) {
            lijevo();
        }
        else if (serialData == 'F' && sonar8.ping_cm() == 0 ||
serialData == 'F' && sonar10.ping_cm() == 0) {
            desno();
        }
        if (serialData == 'S')
            stani();
        if (serialData == 'L')
            lijevo();
        if (serialData == 'R')
            desno();
        if (serialData == 'Q')
            naprijed_lijevo();
        if (serialData == 'E')
            naprijed_desno();
        if (serialData == 'Z')
            nazad_lijevo();
        if (serialData == 'C')
            nazad_desno();
    }
}
```

Izvor: Nenad Vagan.