

# Документација за проект по предметот Основи на роботика

## „ГАРДЕНИНО“

### Вовед

Овој проект претставува изработка на робот, подвижна количка управувана со инфрацрвен далечински управувач, која дополнително користи ултразвучни сензори за детекција на пречки, LCD дисплеј за прикажување на податоци добиени од сензорите, DHT11 сензор за мерење на влажност и температура на воздух, сензор за мерење на влажност на почва, потопна пумпа за наводнување и сеалка за семки која работи со помош на серво мотор.

### Цел на проектот

Целта на овој проект е да се изработи подвижна количка управувана со далечински управувач, која може да дава прецизни податоци на земјоделецот за временските услови и истата може да се корисити за садење на различни земјоделски култури. Проектот е инспириран од големиот неискористен земјоделски потенцијал на Македонија. Дополнително, количката е опремена со ултразвучни сензори за препознавање на пречки, што ја прави посигурна и практична при движење, LCD дисплеј на кој во реално време ги прикажува податоците за температура на воздух и влажност добиени од сензорите.

### Потребни делови:

- Arduino Uno
- LCD дисплеј со I2C интерфејс(<https://share.temu.com/hbtpmBZA2HB>)
- Jumper жици
- Прекинувач
- Инфрацрвен модул за далечински управувач со инфраред сензор
- 2 ултразвучни сензори за далечина
- 2 литиумски батерии од 3.7 V
- Држач за 2 батерии(<https://share.temu.com/Ow8fRJd7lgB>)
- Breadboard плоча за поврзување со Ардуино
- 4 DC моторчиња со тркало(<https://share.temu.com/ATkUpRN7lqB>)
- L298N контролер за DC Мотор(<https://share.temu.com/RddS49Pfi1B>)
- DHT11 сензор за влажност на воздухот и температура
- Сензор за мерење на влажност на почвата
- Пумпа за вода(<https://share.temu.com/TcQrwCDXQMB>)
- Серво мотори за сеалка
- Батерија за напојување 9V

- 5V/12V Пеле модул(<https://share.temu.com/8RMV72DiwgB>)
- Инка
- Серво мотор
- Останати помошни делови(дистанцери, подлоги, контејнери инт)

## **Зошто се користат овие делови**

### **Arduino Uno**

Главниот микроконтролер кој го координира целиот систем. Тој прима сигнали од далечинскиот управувач и сензорите, и врз основа на тоа праќа команди до моторите и LCD дисплејот.

### **LCD дисплеј со I2C интерфејс**

Служи за прикажување на моменталната команда на количката. I2C интерфејсот овозможува полесно поврзување со Ардуино бидејќи бара помалку жици.

### **Jumper жици**

Овозможуваат електрично поврзување на сите компоненти со Arduino плочката и breadboard-от.

### **Прекинувач**

За вклучување и исклучување на количката на едноставен начин.

### **Инфрацрвен модул со IR сензор**

Го прима сигналот од далечинскиот управувач и го претвора во дигитална вредност што Arduino ја разбира. Благодарение на него можеме со едноставни копчиња (2, 4, 5, 6, 8) да ја контролираме количката.

### **2 ултразвучни сензори**

Служат за мерење на растојание до некоја пречка. Со нив количката може да „види“ ако има објект пред неа и да избегне судири со тоа што ќе застане.

### **2 литиумски батерии од 3.7V + држач**

Обезбедуваат напојување на целиот систем. Двете батерии поврзани даваат доволно напон и струја за да ги напојуваат и Arduino плочката и моторите.

### **Breadboard плоча**

Овозможува поврзување на сите компоненти без потреба од лемење.

### **4 DC моторчиња со тркало**

Овозможуваат движење на количката. Со комбинација на нивното вртење количката може да оди напред, назад или да врти во лево/десно.

### **L298N мотор контролер**

Посебен модул за управување со DC моторите. Тој ја регулира насоката и брзината на моторите врз основа на сигналите добиени од Arduino.

### **DHT11 сензор**

Релативно евтин сензор кој дава податоци за температурата и влажноста на воздухот со точност од  $\pm 2$ .

### **Потопна пумпа за вода**

Потопна пумпа за вода која повлекува вода од посебен контејнер и ја наводнува земјоделската култура.

### **Батерија за напојување од 9V**

Посебна батерија за напојување на пумпата бидејќи батериите од 3.7V не се доволни.

### **5V/12V реле модул**

Реле модул кој се користи за гасење и палење на потопната пумпа за наводнување.

### **Сензор за мерење на влажност на почва**

Ја мери влажноста на земјата со помош на метални електроди и враќа вредност до Ардуиното.

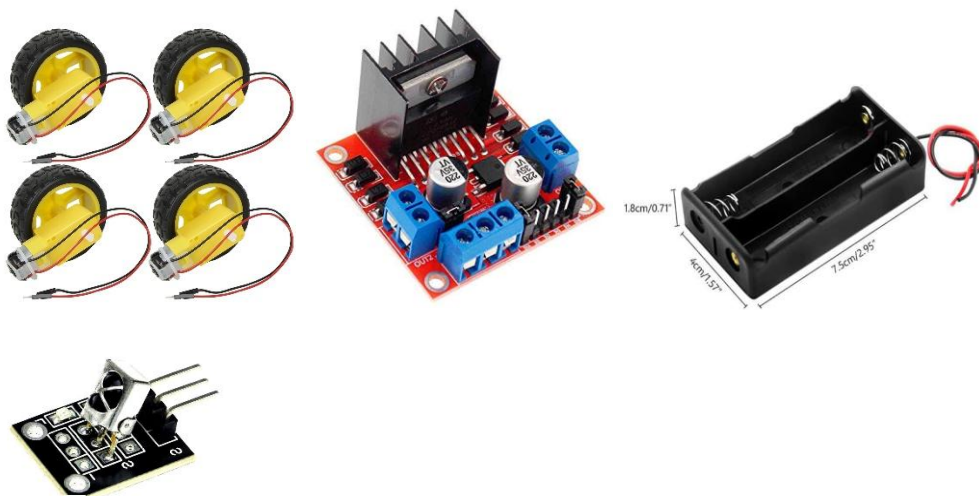
### **Инка**

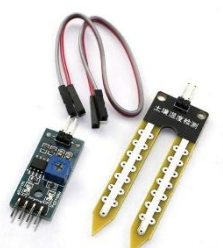
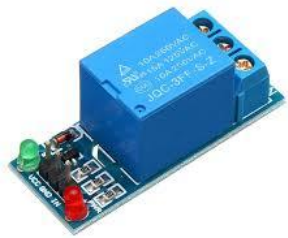
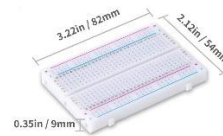
Инка која ги чува семките.

### **Серво мотор**

Се користи за отварање и затворање на инката со семки. Со неговото поместување се испуштаат семки во почвата.

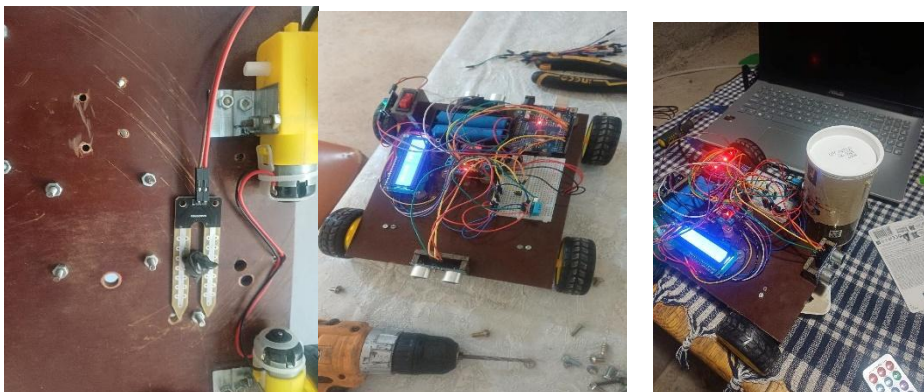
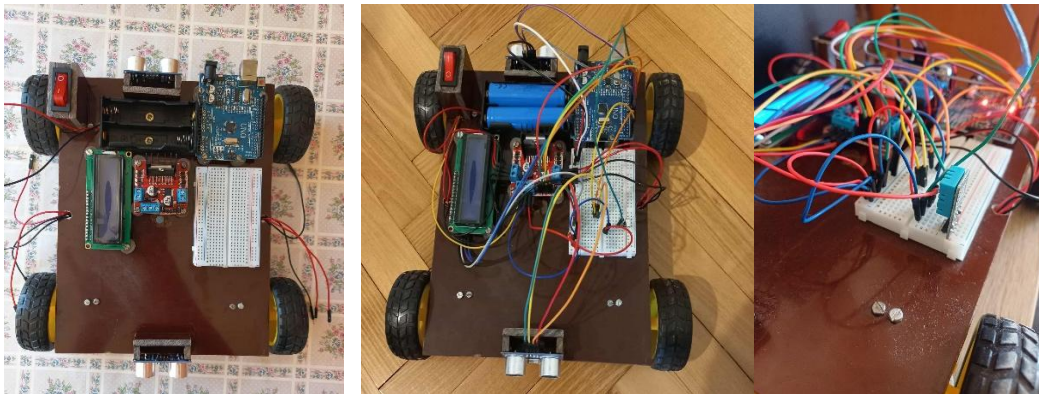
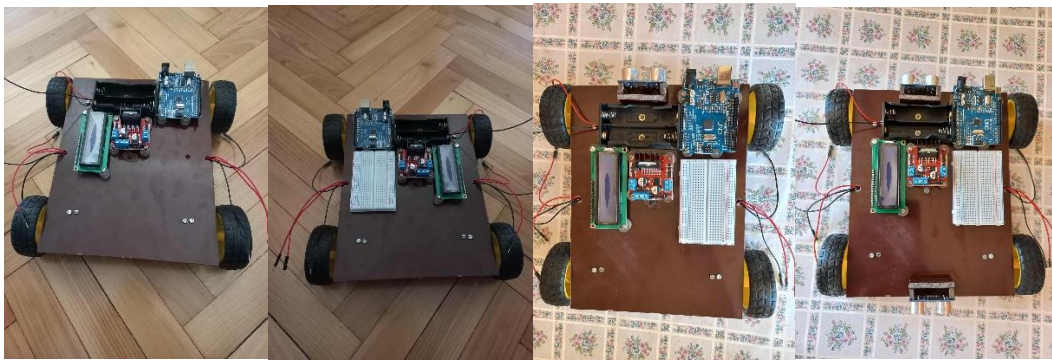
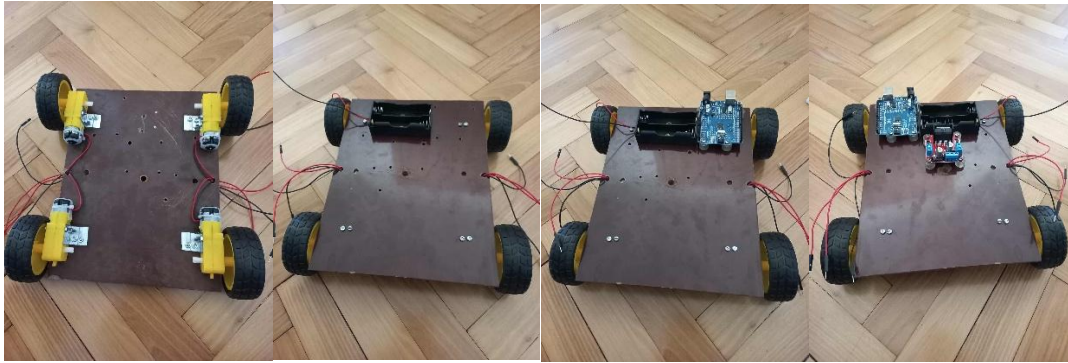
### **Слики од деловите за проектот:**



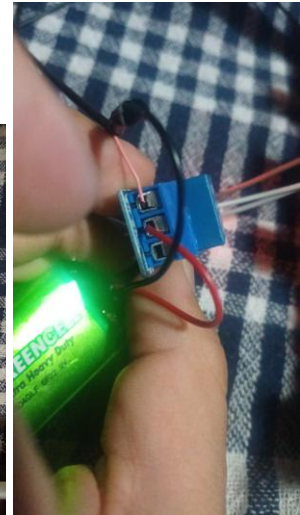
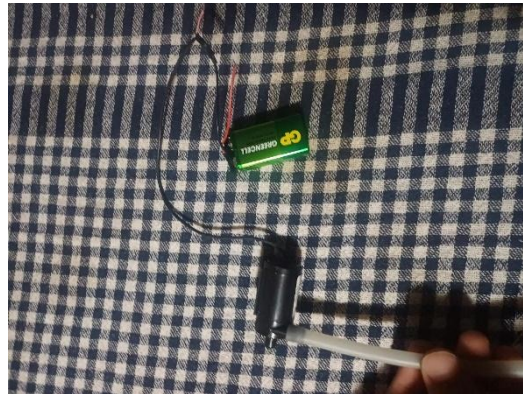




## Слики од процесот на изработка:







## Начин на изработка

Роботот беше изработен за една седмица. Прво беа поврзани моторчињата и тркалата на платформата. Потоа се додадоа батериите, ардуиното и L298 контролерот кој овозможува контрола на насоката на електро моторчињата кои го придвижуваат роботот. Наредно ЛЦД дисплејот се лемеше на I2C интерфејс. Контролата на роботот се вршеше со далечинско, па така со помош на инфрацрвен модул со „IR“ сензор од далечинско се праќаа сигнали за работа на роботот. Откако главните компоненти беа на место наредно беше да се наместат различните сензори и да се додадат неговите „алатки“. Прво ја додадовме пумпа која ја поврзавме со помош на реле. Ова реле овозможува по желба да ја пуштаме и гасиме пумпата. Сеалката беше додадена наредна за да се овозможи роботот да сее семки. Од сензори беа имплементирани сензорите кои сметавме дека најмногу ќе помогнат на земјоделците. Тоа беа сензори за мерење на температура и влажност на воздух и на крај сензор за влажност на почва. Најголеми потешкотии беа да се најде соодветна платформа на која ќе стојат деловите и соодветни дистанцери кои ќе ги поткреваат деловите да не дојде до некаков спој. За подлога искористивме голема пластика која ја

најдовме во подрум, а дистанцери се користеа од пластика и вишок од матична плоча. Битно е да се напомене дека за поврзување на сите делови се користеше „breadboard“ бидејќи само со помош на жици и без лемење деловите и сензорите можат лено да се стават во функција. Исто така овој начин овозможува лесно реискористување на деловите за други проекти. При пишување на кодот имавме малку потешкотии бидејќи требаше да истражуваме за различните начини на кои сензорите ги праќаат податоците.

## **Предности и маани на роботот**

Од предности, нашиот робот наречен „Гарденино“, овозможува приказ на сите параметри кои на еден земјоделец или градинат би му помогнале да насади едно растение. Како на пример температурата и влажноста на воздухот кои играат главна улога во ’ртењето на семето. Доколку градинарот се двоуми за полевање на културите, со помош на сензорот за влажност на почвата може да измери дали е потребна вода или не. Мааните на роботот се тоа што контејнерот со вода би можел лесно да ја оштети електрониката доколку капка би стигнала до жиците. Исто така со помош на хидраулика и тегови би можело да се направи сензорот за влажност на почвата автоматски да се забодува во земјата.

**Линк до видео како работи:** <https://youtu.be/x4z1MsV4iMM>

## **Идни подобрувања на роботот**

На „Гарденино“ би можело да се додаде вештачка интелигенција која автоматски би работела автономно. За да се овозможи и човекот да има увид во работата на роботот и вештачката интелигенција би сакале да додадеме и далечинско управување на роботот. Ова би овозможило човекот од топлината на својот дом со помош на камера вградена на роботот да ја надгледува состојбата на своите поседи. Една од насоките во кои би требале да размислуваме е и садењето на плодови со ЦПС. Доколку се маркира позицијата на секој плод со помош на ЦПС, со вештачката интелигенција би можеле да ја следиме состојбата на секое стебленце и да надгледуваме дали е болно, му фали некој минерал итн.

## Ардуино код за роботот:

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <IRremote.h>
#include <string.h>
#include <DHT.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo Servo1;

#define echoPin1 2
#define trigPin1 3
#define echoPin2 4
#define trigPin2 5

int DC1 = 6;
int DC2 = 7;
int DC3 = 8;
int DC4 = 9;
int RemotePin = 10;
int servoPin = 11;
int relay_pin = 12;

int MOISTURE_PIN = A2;
#define DHT_PIN A3

#define DHT_TYPE DHT11
DHT dht(DHT_PIN, DHT_TYPE);
int temperature=0;
int humidity=0;
unsigned long lastDHT = 0;
const unsigned long DHT_INTERVAL = 2000;

int moisture_percent=0;
int soil_moisture = 0;
int dry_value = 940;
int wet_value = 300;

long lastCommand = 0;
unsigned long lastSignalTime = 0;
const unsigned long RELEASE_TIMEOUT = 100;

int distance1=0;
int distance2=0;

String lcd_message;

enum BUTTON_PRESSED{
    NUM_ONE = 0xC,
    NUM_TWO = 0x18,
    NUM_THREE = 0x5E,
    NUM_FOUR = 0x8,
    NUM_FIVE = 0x1C,
    NUM_SIX = 0x5A,
    NUM_SEVEN = 0x42,
    NUM_EIGHT = 0x52,
    NUM_NINE=0x4a
};

void setup() {
    Serial.begin(9600);
    Servo.attach(servoPin);
    IrReceiver.begin(RemotePin);
    dht.begin();
    lcd.init();
    lcd.backlight();

    pinMode(relay_pin,OUTPUT);
    digitalWrite(relay_pin, HIGH);

    pinMode(DC1, OUTPUT);
    pinMode(DC2, OUTPUT);
    pinMode(DC3, OUTPUT);
    pinMode(DC4, OUTPUT);

    pinMode(trigPin1, OUTPUT);
    pinMode(echoPin1, INPUT);

    pinMode(trigPin2, OUTPUT);
    pinMode(echoPin2, INPUT);

    lcd_message = "Welcome farmer";
    print_message(lcd_message);
}

long lastLcdCommand = -1;

void loop() {
    input();
    detect_obstacle();

    unsigned long now = millis();
    if (now - lastDHT >= DHT_INTERVAL) {
        lastDHT = now;
        SOIL_MOISTURE_READ();
        DHT_READ();

        if (lastCommand != lastLcdCommand) {
            lastLcdCommand = lastCommand;

            lcd.clear();
            if (lastCommand==NUM_ONE){
                DISPLAY_READINGS("Temperature: ", temperature, 0);
                DISPLAY_READINGS("Humidity: ", humidity, 1);
            } else if (lastCommand==NUM_THREE){
                DISPLAY_READINGS("Moistures: ",moisture_percent,0);
            } else if (lastCommand==NUM_SEVEN){
                lcd_message = "Watering";
                print_message(lcd_message);
                digitalWrite(relay_pin, LOW);
            } else{
                lcd_message = "Welcome farmer";
                print_message(lcd_message);
                digitalWrite(relay_pin, HIGH);
            }
        }
    }
}
```



```
void SOIL_MOISTURE_READ(){
    soil_moisture = analogRead(MOISTURE_PIN);
    moisture_percent = map(soil_moisture, dry_value, wet_value, 0, 100);
    if (moisture_percent < 0) moisture_percent = 0;
    if (moisture_percent > 100) moisture_percent = 100;
}

void DHT_READ(){
    temperature = dht.readTemperature();
    humidity = dht.readHumidity();
}

void DISPLAY_READINGS(String text, int reading, int cursor){
    lcd.setCursor(0, cursor);
    lcd.print(text);
    lcd.print(reading);
}

void print_message(String message) {
    lcd.clear();
    if (message.length() <= 16) {
        lcd.print(message);
    } else {
        delay(50);
        for (int i = 0; i <= message.length() - 16; i++) {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print(message.substring(i, i + 16));
            delay(300);
        }
        delay(700);
    }
}
```

```

void detect_obstacle(){
    distance1 = measureDistance(trigPin1, echoPin1);
    distance2 = measureDistance(trigPin2, echoPin2);

    if ((lastCommand == NUM_TWO || lastCommand == 0xFFFFFFFF) && distance2 < 30) {
        move("stop");
    }

    if ((lastCommand == NUM_EIGHT || lastCommand == 0xFFFFFFFF) && distance1 < 30) {
        move("stop");
    }
}

void input() {
    if (IrReceiver.decode()) {
        long value = IrReceiver.decodedIRData.command;
        Serial.println(value, HEX);

        if (value == 0xFFFFFFFF) {
            value = lastCommand;
        } else if (value != 0) {
            lastCommand = value;
        }

        if (value != 0) {
            switch(value){
                case NUM_ONE:
                    Serial.print("TEST");
                    break;

                case NUM_TWO:
                    move("forward");
                    break;

                case NUM_EIGHT:
                    move("backward");
                    break;

                case NUM_FOUR:
                    move("left");
                    break;

                case NUM_SIX:
                    move("right");
                    break;

                case NUM_NINE:
                    seed();
                    break;

                default:
                    move("stop");
                    break;
            }
        }

        lastSignalTime = millis();
        IrReceiver.resume();
    }

    if (millis() - lastSignalTime > RELEASE_TIMEOUT) {
        move("stop");
        lastCommand = 0;
    }
}

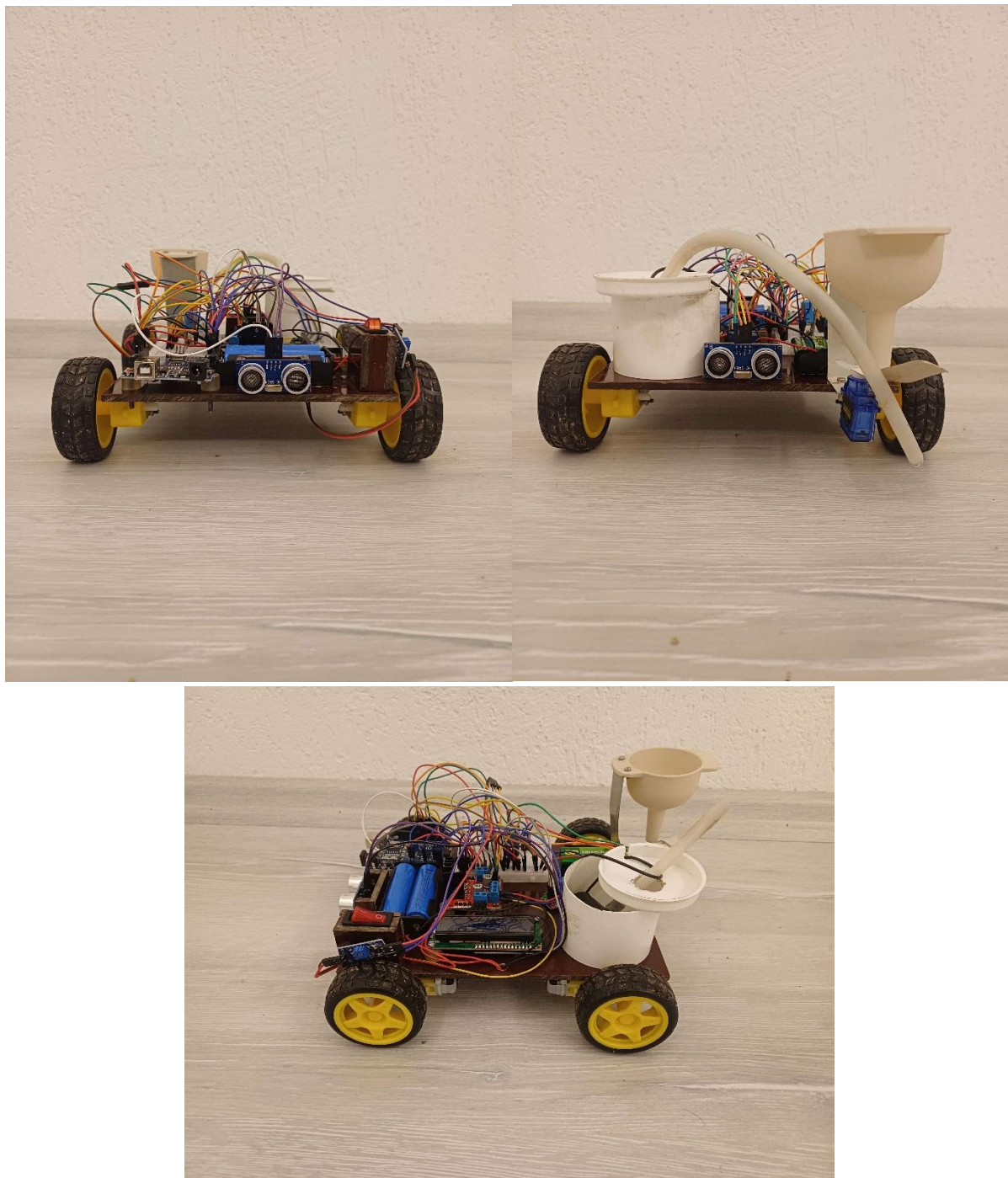
void move(String dir) {
    if (dir == "forward") {
        digitalWrite(DC1, HIGH); digitalWrite(DC2, LOW);
        digitalWrite(DC3, HIGH); digitalWrite(DC4, LOW);
    }
    else if (dir == "backward") {
        digitalWrite(DC1, LOW); digitalWrite(DC2, HIGH);
        digitalWrite(DC3, LOW); digitalWrite(DC4, HIGH);
    }
    else if (dir == "left") {
        digitalWrite(DC1, LOW); digitalWrite(DC2, HIGH);
        digitalWrite(DC3, HIGH); digitalWrite(DC4, LOW);
    }
    else if (dir == "right") {
        digitalWrite(DC1, HIGH); digitalWrite(DC2, LOW);
        digitalWrite(DC3, LOW); digitalWrite(DC4, HIGH);
    }
    else{
        digitalWrite(DC1, LOW); digitalWrite(DC2, LOW);
        digitalWrite(DC3, LOW); digitalWrite(DC4, LOW);
    }
}

void seed(){
    Servo1.write(170);
    delay(200);
    Servo1.write(100);
    delay(1000);
    Servo1.write(170);
}

long measureDistance(int trigPin, int echoPin) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long duration = pulseIn(echoPin, HIGH);
    return duration * 0.034 / 2;
}

```

## Завршен продукт



Изработиле:

Елена Николовска-225004

Христијан Мијалески Спасески-223211