



Univerzitet u Beogradu  
Elektrotehnički fakultet

Studentski projekat iz predmeta Arhitektura IK sistema  
2022/2023

# Kontrola dva servo motora preko Bluetooth modula

Studenti: Minja Milovanović 2020/0126, Jovana Marić  
2020/0144, Mihajlo Ćulibrk 2020/0156, Emilija Lalković  
2020/0518

Mentor: dr Nikola Basta, docent

## Sadržaj

1. Uvod	3
2. Komponente	3
3. Povezivanje komponenti	6
4. Implementacija koda	14
5. Zaključak	20
6. Literatura	20

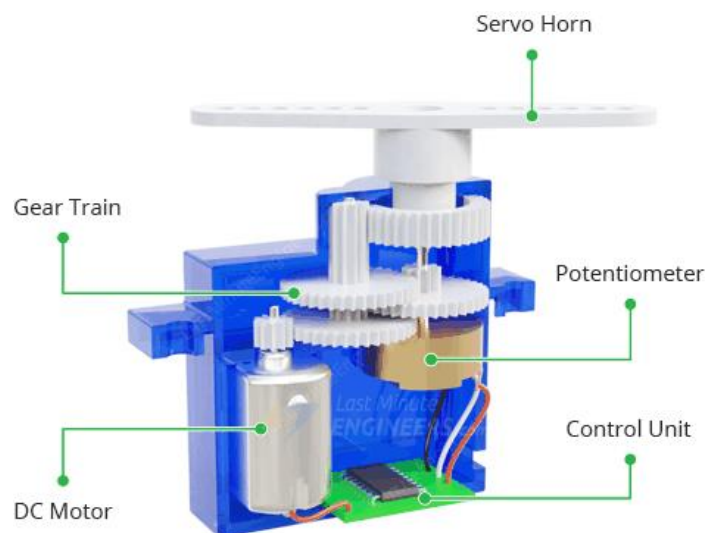
## 1. Uvod

### 1.1 Kratak opis projekta

Naš zadatak podrazumeva realizaciju jednostavnog uređaja kojim se upravlja dvama servo motorima, kao i proširenje funkcionalnosti uređaja tako da se omogući bežična kontrola preko Bluetooth modula. Platforma koju ćemo koristiti je Arduino MEGA 2560 R3.

### 1.2 Uopšteno o servo motorima

Servo motori su aktuatori koji omogućavaju preciznu kontrolu položaja ili brzine. Sastoji se od odgovarajućeg motora koji je povezan sa senzorom koji mu šalje odgovarajuće povratne informacije. U našem slučaju to su motor i rotacioni enkoder koji u zavisnosti od ulaza održavaju konstantan ugao. Naš servo motor, kao i mnogi drugi, se kontroliše dužinom impulsa. U zavisnosti od procenta dužine impulsa u odnosu na maksimalnu dužinu koju definiše proizvođač on podešava svoj ugao u opsegu  $\sim(0\ 180)$ . Na primer, ovakav servo motor se takođe može koristiti za modelarske avione sa daljinskom komandom, gde bi se koristio da podešava ugao zakrilca.



Slika 1. Servo motor.

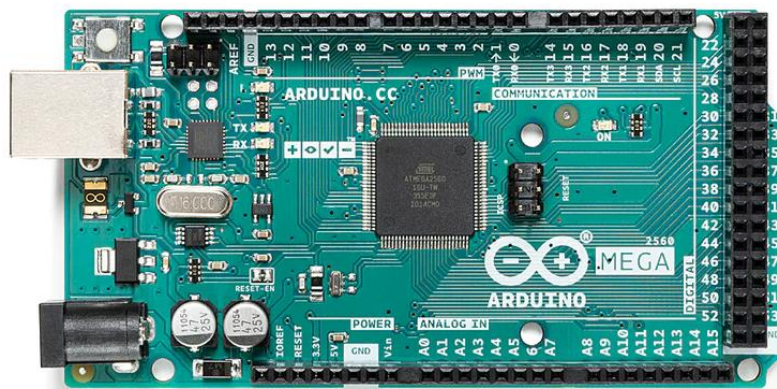
### 1.3 Uopšteno o mikrokontrolerima

Mikrokontroler je mali računar koji se nalazi na čipu. Sadrži jedan ili više CPU i memoriju, kao i neophodne programabilne ulazne/izlazne periferije. Zbog manje veličine i cene u poređenju sa dizajnom koji koristi odvojeni mikroprocesor, memoriju i ulazne/izlazne periferije, mikrokontroleri su znatno ekonomičniji za digitalnu kontrolu više uređaja i procesa. Mikrokontroleri su programabilni, što znači da se mogu prilagoditi za izvršenje specifičnih radnji, kao što je u našem projektu cilj omogućiti upravljanje servo motorima. Programski jezik koji se koristi bazira se na jezicima C, C++ i asemblerskom jeziku.

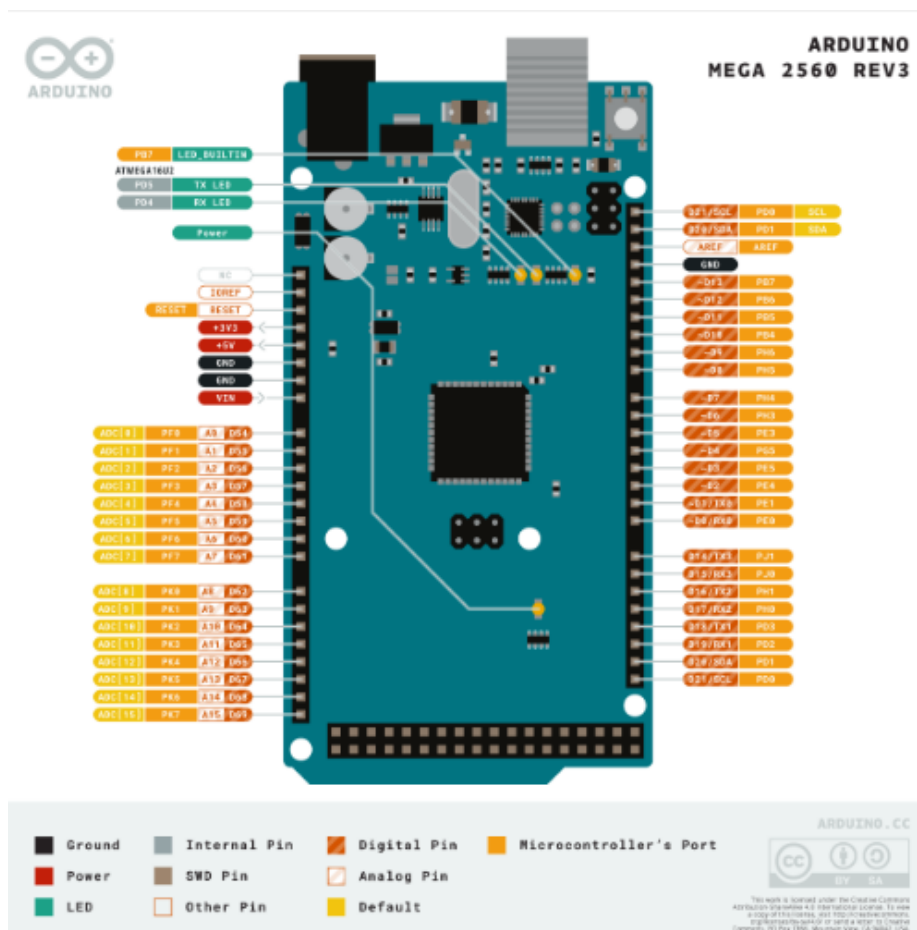
## 2. Komponente

### 2.1 Arduino Mega 2560 Rev3

Arduino Mega je ploča mikrokontrolera zasnovana na ATmega2560. Ima 54 digitalna ulazno/izlazna pina, 16 analognih ulaza, 4 hardverska serijska porta, kristalni oscilator od 16MHz, USB konekciju, priključak za napajanje, ISCP zaglavlje i dugme za resetovanje. Od 54 digitalna I/O pina, 15 su PWM pinovi. Ima dva pina za napajanje spoljnih komponenti (5V i 3,3V) i dva za uzemljenje (ground).



Slika 2.Arduino Mega 2560 Rev3.



Slika 3.Arduino Mega 2560 Rev3 - raspored pinova.

## 2.2 Arduino servo motor

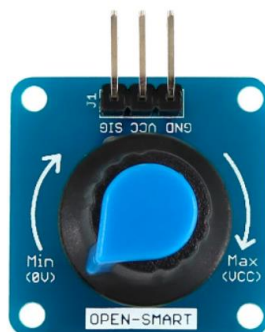
Servo motori imaju tri žice: napajanje, uzemljenje i signal. Žica za napajanje je crvena i treba da bude povezana na 5V pin na Arduino ploči. Žica za uzemljenje je crna ili braon i povezuje se sa uzemljenjem na ploči. Signalni pin je žut ili narandžast i treba ga povezati sa PWM pinom na ploči.



Slika 4.Arduino servo motor 9g, SG90.

### 2.3 Arduino rotacioni potencijometar

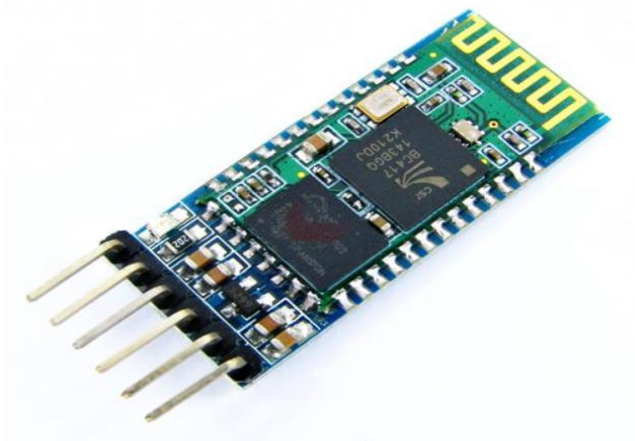
Potencijometar se može koristiti kao senzor ugla rotacije koji stvara analogni izlaz između 0 i Vcc na svom digitalnom pinu. Ugaoni opseg je 280 stepeni sa linearnom promenom vrednosti.



Slika 5.Arduino rotacioni potencijometar.

### 2.4 Arduino Bluetooth modul

Omogućava Arduino da šalje i prima TTL podatke preko Bluetooth tehnologije bez povezivanja serijskog kabla na računar. Može da se koristi u master i slave konfiguraciji. Ima šest pinova: Key/Enable, Vcc, Ground, TXD(predaja), RXD(prijem) i State. Ukoliko Enable ima vrednost high, modul radi u režimu naredbe(command mode), odnosno koristi AT naredbe za podešavanje modula; u suprotnom je u režimu prenosa podataka. Napon napajanja je 5V ili 3,3V. TXD/RXD je serijska predaja/prijem podataka. State govori o tome da li je modul povezan ili ne.



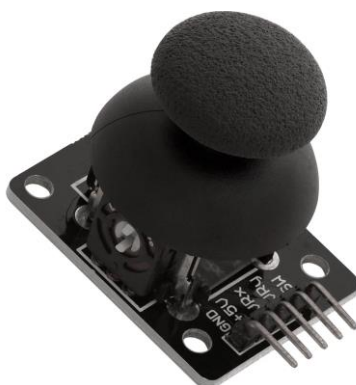
Slika 6.Bluetooth modul HC-05.



Slika 7 .Bluetooth modul HC-05 - raspored pinova.

## 2.4 Arduino džojstik

Džojstik se sastoji od dva potenciometra koja su normalna i tastera. Pruža dve analogne vrednosti u rasponu od 0 do 1023 koje odgovaraju horizontalnom (X) i vertikalnom (Y) položaju.



Slika 8.Arduino joystick.

## 3. Povezivanje komponenti

### 3.1 Upravljanje dvama servo motorima

Za povezivanje svih komponenti koristili smo kratkospojnice tipa F/F (ženski priključak - ženski priključak) i M/M (muški priključak - muški priključak).



Slika 9. kratkospojnice tipa F/F.



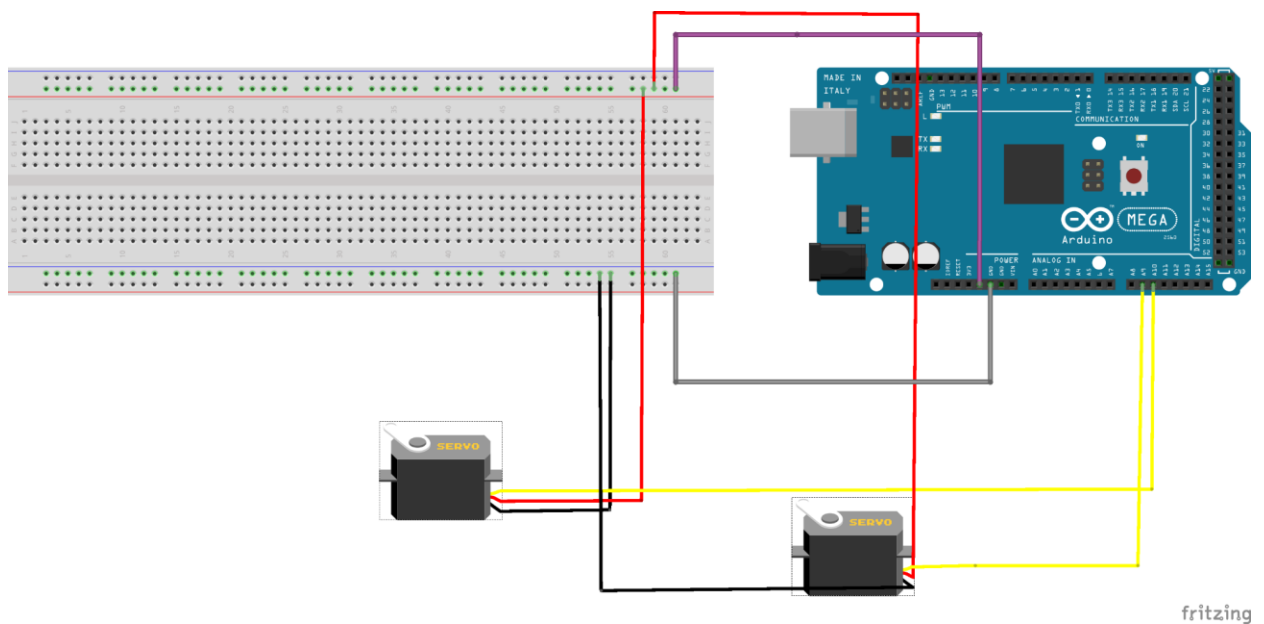
Slika 10. kratkospojnice tipa M/M.

Takođe je korišćena prototipska ploča sa slike.



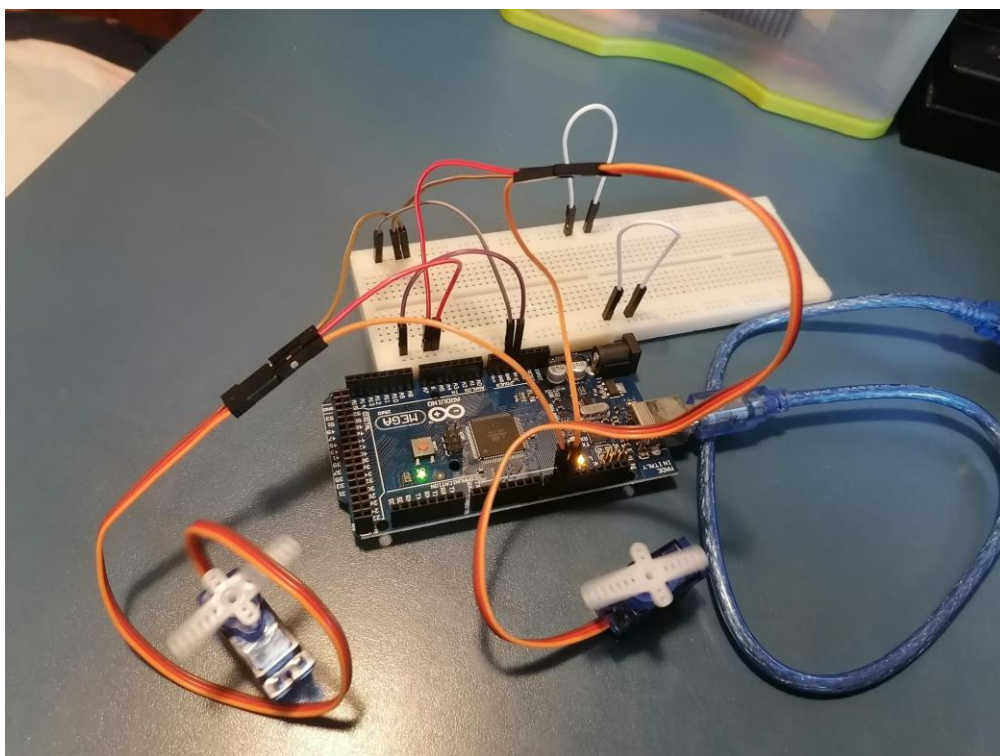
Slika 11. prototipska ploča (protoboard).

Najpre smo povezali oba servo motora sa Arduinoom i komande zadali programski tako da se motori okreću od 0 do 180 stepeni.



Slika 12. Šema povezivanja dva servo motora sa Arduinoom za programsko upravljanje.





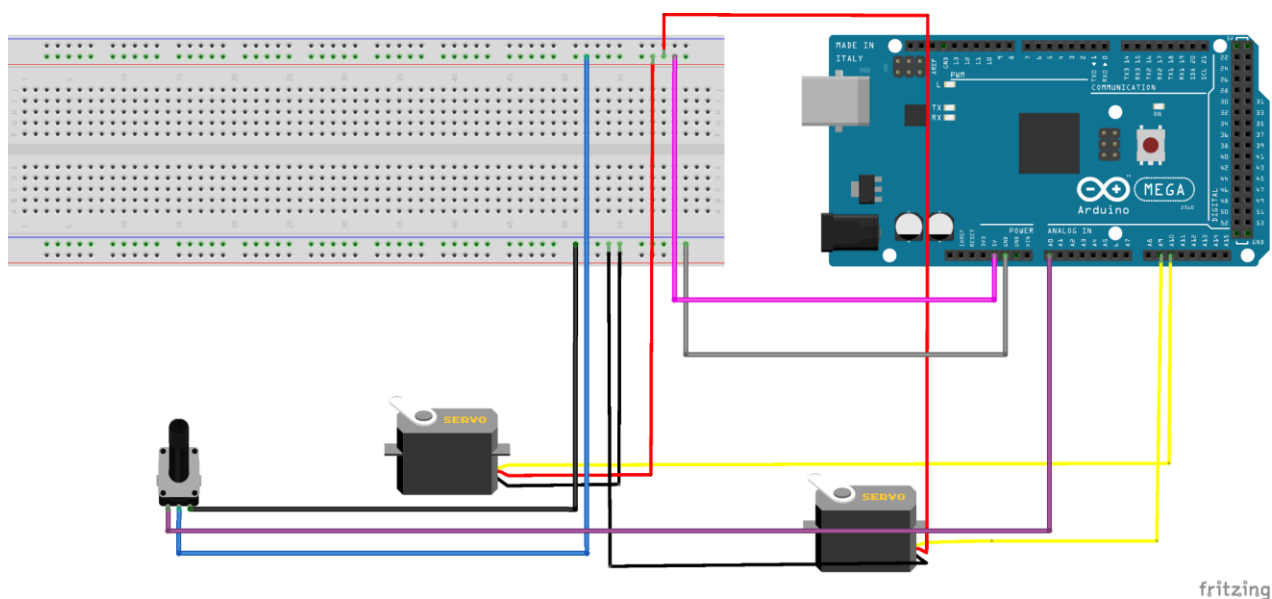
Slika 13. Povezivanje dva servo motora sa Arduniom za programsko upravljanje.

Tabela 1. Povezivanje pinova servo motora.

Servo motor	Funkcija pina	Arduino MEGA 2560
GND	Uzemljenje	GND
VCC	Napajanje	5V
CONTROL	Signal	A9/A10

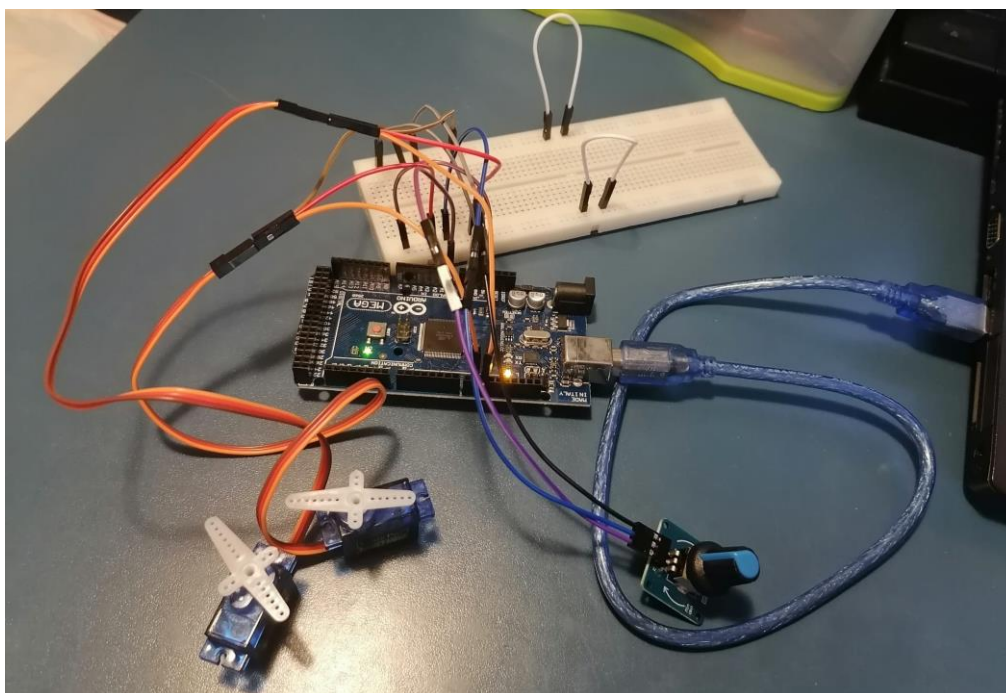
Zatim smo povezali servo motore tako da omogućimo upravljanje motorima korišćenjem potenciometra, tj. zadavanje komandi za kretanje korišćenjem potenciometra koji mapira otpornosti od 0 do 1023 u uglove od 0 do 180 stepeni.





Slika 14. Šema dva servo motora povezana sa Arduinom i pločom, finalno rešenje.

Napomena: Pošto u fritzing-ovom odeljku "Parts" ne postoji potencijometar koji smo koristili (Arduino rotacioni potencijometar), korišćen je rotacioni potencijometar koji je bio dostupan.



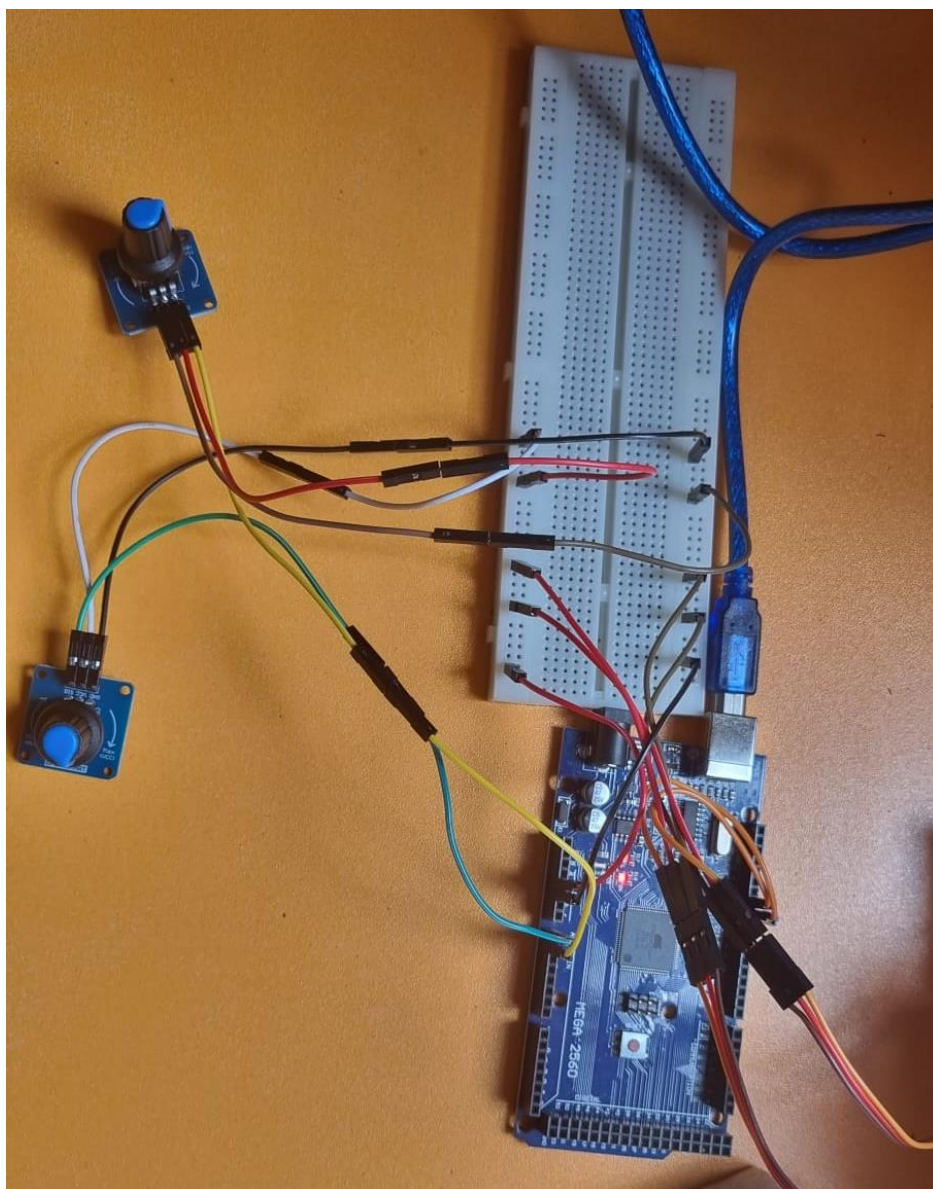
Slika 15. Dva servo motora povezana sa Arduinom i pločom, finalno rešenje.

Tabela 2. Povezivanje pinova potencijometra.

Potencijometar	Funkcija pina	Arduino MEGA 2560
GND	Uzemljenje	GND
VCC	Napajanje	5V
CONTROL	Signal	A0

The diagram illustrates the hardware setup for a robotic arm. An Arduino Mega 2560 is connected to a breadboard. The breadboard contains a 5V regulator and a 10k pull-down resistor. Wires connect the Arduino's 5V, GND, and digital pins to the breadboard and four servos. The servos are labeled 'SERVO' and are connected to specific digital pins on the Arduino.

10



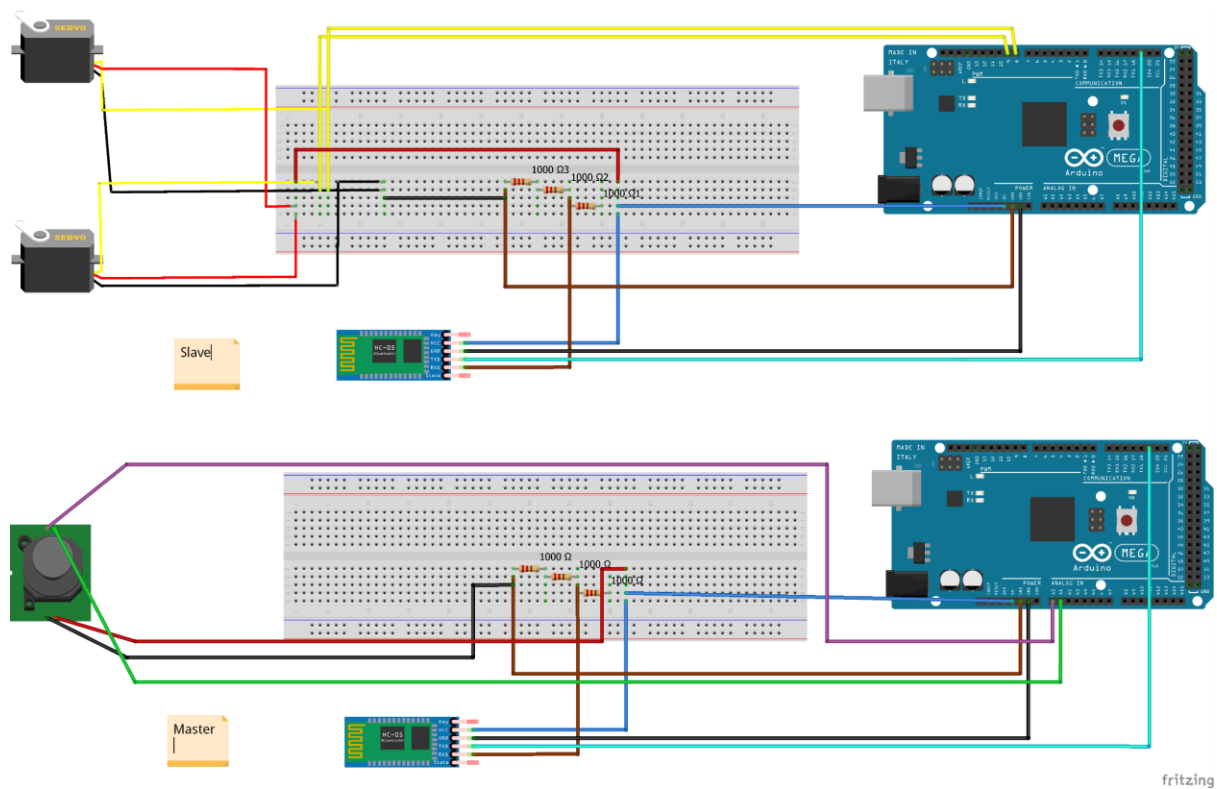
Slika 18. Komponente za dizalicu.

Tabela 3. Povezivanje pinova potenciometra.

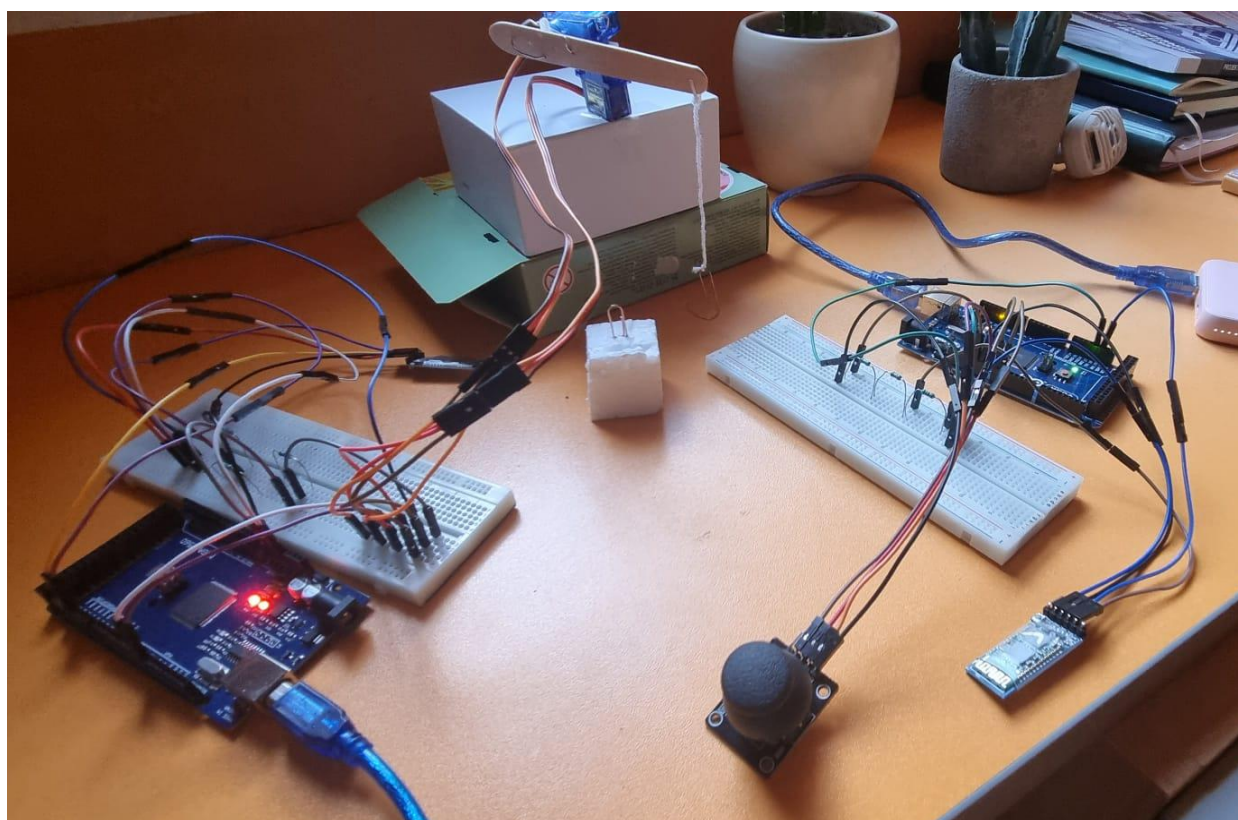
Potenciometar	Funkcija pina	Arduino MEGA 2560
GND	Uzemljenje	GND
VCC	Napajanje	5V
CONTROL	Signal	A0 / A1

### 3.2 Upravljanje servo motorima preko Bluetootha

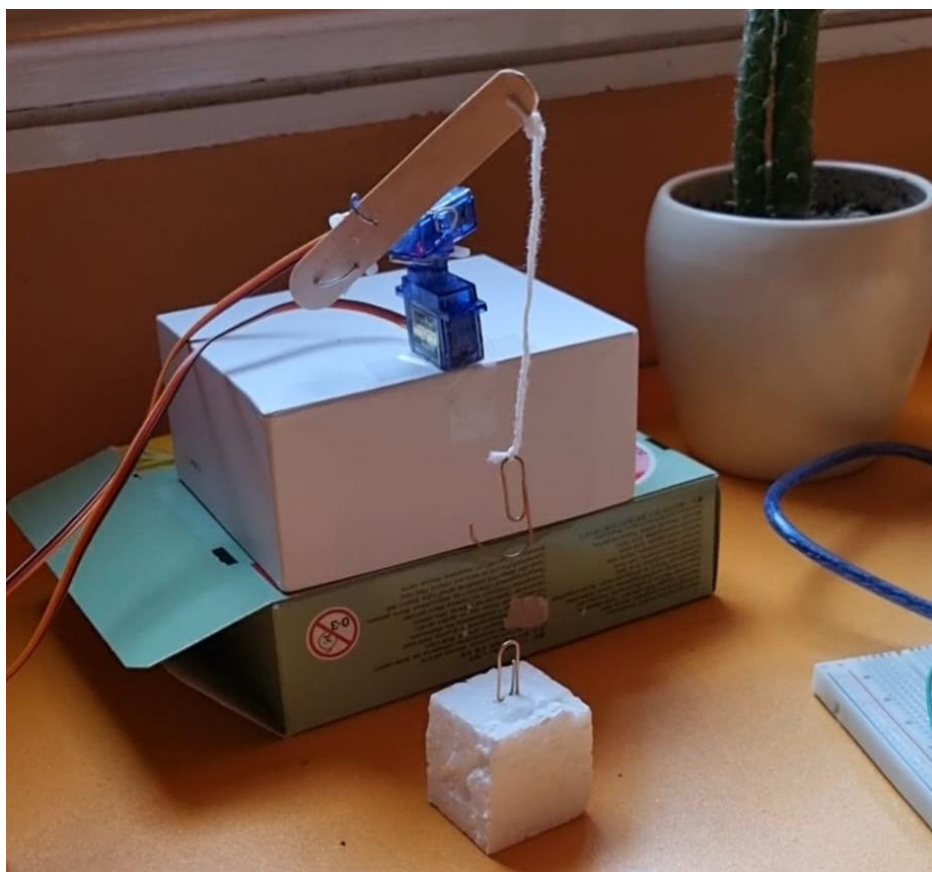




Slika 19. Upravljanje dvama servo motorima korišćenjem Bluetootha i džojstika (šema).



Slika 20. Upravljanje dvama servo motorima korišćenjem Bluetootha i džojstika.



Slika 21. Dizalica od dva servo motora.

Tabela 4. povezivanje džojstika.

Džojstik	Funkcija pina	Arduino MEGA 2560
GND	Uzemljenje	GND
+5V	Napajanje	5V
VRX	Menjanje horizontalnog položaja	A1
VRY	Menjanje vertikalnog položaja	A0

Tabela 5. povezivanje Bluetooth modula.

Bluetooth HC-05	Funkcija pina	Arduino MEGA 2560
GND	Uzemljenje	GND
VCC	Napajanje	5V
RXD	Serijski prijem podataka	TX1
TXD	Serijsko slanje podataka	RX1

## 4. Implementacija koda

### 4.1 Programsko okruženje Arduino IDE

Arduino programski jezik zasnovan je na C/C++ jeziku. Dve funkcije se koriste u svakom kodu u programskom okruženju Arduino IDE i to su:

- void setup() - izvršava se samo jednom i koristi se za inicijalizaciju početnih vrednosti
- void loop() - izvršava se sve vreme, dok se ne isključi napajanje

Void označava da funkcija ništa ne vraća nakon izvršenja.

### 4.2 Upravljanje dvama servo motorima

#### 4.2.1 Korak 1

```
#include <Servo.h> //servo biblioteka

Servo servo_obj1; //1. servo motor
Servo servo_obj2; //2. servo motor
int ugao = 0; //za čuvanje vrednosti ugla
int pinServo1 = 9; //pin iz prvog servo motora
int pinServo2 = 10; //pin iz drugog servo motora
void setup() {
    servo_obj1.attach(pinServo1); //deklarisanje 2 tipa motora
    servo_obj2.attach(pinServo2);
}
void loop() {
    for (ugao = 0; ugao <= 180; ugao += 2) {
        servo_obj1.write(ugao); //servo motori se pomeraju za po 2 stepena
        servo_obj2.write(ugao); //pomeraju se u rasponu od 0 do 180
        delay(10); //čekanje
    }
    for (ugao = 180; ugao >= 0; ugao -= 2) {
        servo_obj1.write(ugao); //vraćanje od 180 do 0 sa korakom -2
        servo_obj2.write(ugao); //svaki korak ugla se upiše u oba motora
        delay(10);
    }
}
```

#### 4.2.2 Korak 2

```
#include <Servo.h> //servo biblioteka
Servo servo_obj1; //1. servo motor
```

```

Servo servo_obj2; //2. servo motor
int pinP =A0; //pin iz potencijometra
int x = 0; //pocetne promenljive, imaju vrednost 0
int y = 0; //vrednost se odmah inicijalizuje
int pinServo1 = 9; //pin iz 1. servo motora
int pinServo2 = 10; //pin iz 2. servo motora

void setup() {
    pinMode(pinP, INPUT); //iz A0 pina učitavamo informacije
    servo_obj1.attach(pinServo1); //deklarisanje 2 tipa motora
    servo_obj2.attach(pinServo2);
}

void loop() {
    x=analogRead(pinP); //učitavamo vrednost iz potencijometra u pinP
    y=map(x,0,1023,0,180); //vrednost ugla koju učitavamo u 2 motora
    //x je trenutna vrednost potencijometra
    //0-1023 je range potencijometra
    //0-180 je range servomotora
    //konvertuje vrednosti potencijometra u vrednosti ugla
    servo_obj1.write(y); //vraca vrednost ugla i upisuje u motor
    servo_obj2.write(y); //vraca vrednost ugla i upisuje u motor
    delay(10);
}

```

#### 4.2.3 Korak 3

```

#include <Servo.h> //servo biblioteka
Servo servo_obj1; //1. servo motor
Servo servo_obj2; //2. servo motor
int pinP1 =A0; //inicijalizacija pomoćnih promenljivih
int pinP2 =A1; //učitavaju se analogne vrednosti iz potencijometra
int x1 = 0; //pomoćna promenljiva za upisivanje rezultata
int y1 = 0; //pomoćna promenljiva za upisivanje rezultata
int x2 = 0; //pomoćna promenljiva za upisivanje rezultata
int y2 = 0; //pomoćna promenljiva za upisivanje rezultata
int pinServo1 = 9; //pin iz 1. servo motora
int pinServo2 = 10; //pin iz 2. servo motora

```



```

void setup() {
    pinMode(pinP1, INPUT); //priprema za učitavanje informacija iz pina
    pinMode(pinP2, INPUT); //priprema za učitavanje informacija iz pina
    servo_obj1.attach(pinServo1); //deklarisanje motora
    servo_obj2.attach(pinServo2); //deklarisanje motora
}

void loop() {
    x1=analogRead(pinP1); //u x1 se učitava vrednost 1. potencijometra
    y1=map(x1,0,1023,0,180); //konvertovana vrednost potencijometra u ugao
    x2=analogRead(pinP2); //u x1 se učitava vrednost 1. potencijometra
    y2=map(x2,0,1023,0,180); //konvertovana vrednost potencijometra u ugao
    servo_obj1.write(y1); //vrednost ugla upisana u servo motor
    servo_obj2.write(y2); //servo motor se okrenuo za upisani ugao
    delay(10);
}

```

#### 4.2.4 Bluetooth

##### 4.2.4.1 Master

```

#include <Servo.h>

int x;
int y;

void setup() {
    Serial.begin(115200); //baud rate za serijsku komunikaciju sa
    racunarom
    Serial1.begin(38400); //baud rate za komunikaciju sa slave-om preko
    Bluetootha
}

void loop() {
    x = analogRead(A1); //x-osa džojstika
    y = analogRead(A0); //y-osa džojstika

    char buffer[40];
    sprintf(buffer, "%04d%04d", x+1, y+1); //bafer se uvodi za efikasnije
    slanje poruke

    Serial.println(buffer);
}

```

```

Serial1.println(buffer);

if (Serial1.available()) {      // Ako nešto dođe u Serial1 (pinovi 0
i 1)
    Serial.write(Serial1.read()); // pročitaj i pošalji na Serial
(USB)
}

delay(500);
}

```

#### 4.2.4.2 Slave

```

#include <Servo.h>

String readString, servol, servo2;
Servo myservo1; //kreiranje servo objekta za upravljanje motorom
Servo myservo2;

int bias_x = 90;
int bias_y = 90;
int max_x = 80;
int max_y = 45;

//inicijalizacija vrednosti za x i za y
int x = 0 + bias_x;
int y = 0 + bias_y;
int step = 2; //uvodimo korak kako koordinate ne bi skakale naglo sa
jedne vrednosti na drugu (npr sa maksimalne moguće na 0) jer dizalica
nije previše stabilna

void setup() {
    Serial.begin(115200);
    Serial1.begin(38400);

    myservo1.attach(8); //pin za kontrolu serva
    myservo2.attach(9);
    myservo1.write(x); //podesavanje pozicije serva
    myservo2.write(y);
    Serial.println("two-servo-test-1.0"); // za praćenje učitano

```

```
}
```

```
void loop() {

    while (Serial1.available()) {
        delay(3); //kasnjenje koje omogucava punjenje bafera
        if (Serial1.available() >0) {
            char c = Serial1.read(); //uzima 1 bajt iz bafera

            if (c == '\r') {
                if (readString.length() > 8) {
                    readString.trim();
                    Serial.println(readString);

                    servo1 = readString.substring(0, 4); //uzima prvih 4 karaktera
                    servo2 = readString.substring(4, 8); //uzima drugih 4 karaktera

                    Serial.println(servo1);
                    Serial.println(servo2);

                    int n1 = servo1.toInt();
                    int n2 = servo2.toInt();

                    int new_x=map(n1-1,0,1023,-max_x,max_x) + bias_x;
                    int new_y=map(n2-1,0,1023,-max_y,max_y) + bias_y;

                    Serial.println(new_x); //na Serial Monitor štampa prethodne
pomeraje
                    Serial.println(new_y);

                    while(abs(x - new_x) > 0 || abs(y - new_y) > 0) {
                        if (new_x > x + step)
                            x += step;
                        else if (new_x < x - step)
                            x -= step;
                        else
```

```

        x = new_x;

        if (new_y > y + step)
            y += step;
        else if (new_y < y - step)
            y -= step;
        else
            y = new_y;

        Serial.print(x);
        Serial.print(" ");
        Serial.println(y);

        myservo1.write(x); //podesava poziciju serva
        myservo2.write(y);
        delay(10);
    }

    Serial.println();

}

    readString = "";
} else {
    readString += c; //makes the string readString
}

}

}
}

```

## 5. Zaključak

Arduino uopšteno važi za platformu koja pruža širok spektar mogućnosti za kreiranje raznolikih rešenja za razne oblasti i jako je dobro dokumentovan. Jednostavno korišćenje omogućava velikom broju korisnika koji nemaju nužno znanje o funkcionisanju mikrokontrolera da naprave razne projekte različitih nivoa složenosti. U našem slučaju smo uspeali da napravimo nešto što više liči na igračku nego na konkretan uređaj za neku ozbiljniju radnju, međutim naš cilj nije bilo kreiranje nekog uređaja već demonstracija mogućnosti koje pruža arduino za kontrolu servo motora. Servo motori generalno imaju široku upotrebu i korisni su kada je neophodno omogućiti neke sofisticirane i precizne pokrete. Naša dizalica je grubi model funkcionisanja servo motora i implementacije Bluetooth-a za kontrolu motora, što je u današnje vreme postalo jedan od najčešćih načina za povezivanje uređaja.

## 6. Literatura

- [1] <https://en.wikipedia.org/wiki/Servomotor>
- [2] <https://www.geeksforgeeks.org/microcontroller-and-its-types/>
- [3] <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino>
- [4] <https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05->
- [5] <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [6] <https://docs.arduino.cc/hardware/mega-2560>
- [7] <https://sensorkit.arduino.cc/sensorkit/module/lessons/lesson/03-the-potentiometer>
- [8] <https://docs.arduino.cc/learn/electronics/servo-motors>
- [9] <https://arduinogetstarted.com/tutorials/arduino-joystick>
- [10] Gunter Spanner: Projekti kućne automatizacije za Arduino: upotreba RFID kompleta za učenje, Agencija Eho, Niš, 2018.
- [11] <https://docs.arduino.cc/learn/electronics/servo-motors>