

# IZRADA MAKETE UREĐAJA ZA SORTIRANJE SLATKIŠA PREMA BOJI

---

**Nesek, Luka**

**Undergraduate thesis / Završni rad**

**2019**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:026117>

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

*Download date / Datum preuzimanja:* **2025-04-01**

*Repository / Repozitorij:*

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



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

**Sveučilišni studij**

**IZRADA MAKETE UREĐAJA ZA SORTIRANJE  
SLATKIŠA PREMA BOJI**

**Završni rad**

**Luka Nesek**

**Osijek, 2019.**

# Sadržaj

<b>1. UVOD</b> .....	1
<b>1.1. Zadatak preddiplomskog rada</b> .....	1
<b>2. PRIMJENJENE TEHNOLOGIJE I ALATI</b> .....	2
<b>2.1. Arduino razvojno okruženje</b> .....	2
<b>2.2. Arduino Nano razvojni sustav</b> .....	3
<b>2.3. Mikro servo motor SG90</b> .....	4
<b>2.4. TCS3200 senzor za prepoznavanje boje</b> .....	5
<b>2.5. Šperploča 4mm</b> .....	8
<b>2.6. Laserski rezač CO<sub>2</sub></b> .....	9
<b>2.7. Postupak rezanja</b> .....	10
<b>2.8. Testiranje sustava</b> .....	11
<b>3. REALIZACIJA SUSTAVA</b> .....	12
<b>3.1. Elektronička shema sustava</b> .....	12
<b>3.2. Izrada makete sustava</b> .....	12
<b>3.3. Programiranje sklopovlja</b> .....	14
<b>4. ZAKLJUČAK</b> .....	17
<b>LITERATURA:</b> .....	18
<b>SAŽETAK</b> .....	19
<b>ABSTRACT</b> .....	20
<b>ŽIVOTOPIS</b> .....	21
<b>PRILOG 1) Kompletna shema nacрта konstrukcije sustava</b> .....	22
<b>PRILOG 2) Slika sustava u pogonu</b> .....	23
<b>PRILOG 3) Program za Arduino razvojni sustav</b> .....	25

# 1. UVOD

Ugradbeni računalni sustav (engl. *Embedded system*) je računalni sustav koji objedinjuje računalno sklopovlje, programsku podršku te eventualne dodatne dijelove (poput mehaničkih ili električnih sustava) s ciljem obavljanja neke specifične zadaće. Ugradbeni računalni sustavi su najčešće ograničeni veličinom, računalnim mogućnostima te zahtijevaju nisku potrošnju, stabilan i pouzdan rad, dok u kritičnim sustavima upravljanja prioritet se stavlja na pravovremeno izvršavanje uz što veću preciznost. S obzirom na očekivanja ugradbenih računalnih sustava za rad u stvarnom vremenu zahtijevaju posebnu programsku podršku dok cjenovno ostaju prihvatljivi. Ugradbeni računalni sustavi najrašireniji su u današnjem svijetu te se nalaze u različitim oblicima od kompleksnih čipova, preko mobitela, automobila, tableta, kapsula i drugo. Cilj ovog rada je izraditi funkcionalnu maketu uređaja za sortiranje slatkiša prema boji. Motivaciju za ovaj rad smo pronašli u jednostavnom olakšavanju, preciziranju i pojednostavljivanju sortiranja objekata, u ovom slučaju bombona.

U poglavlju dva smo opisali korištenje primijenjene tehnologije i alata, uz korištenje Arduino okruženja i sustava. Opisali smo ulogu servo motora i ključnog senzora za prepoznavanje boja. Također smo pri izradi koristili laserski rezač za precizno rezanje konstrukcije sustava koju smo napravili od šper ploča, te smo opisali postupak laserskog rezanja toplinom i testirali sustav sortiranjem predviđenih objekata. Zatim smo u trećem poglavlju opisali elektroničku shemu sustava, te objedinjenost računalnog sklopovlja i mehaničkih dijelova, te smo pojasnili programski dio sustava. U zaključku smo iznijeli steknuta saznanja dobivena izradom i testiranjem ove makete.

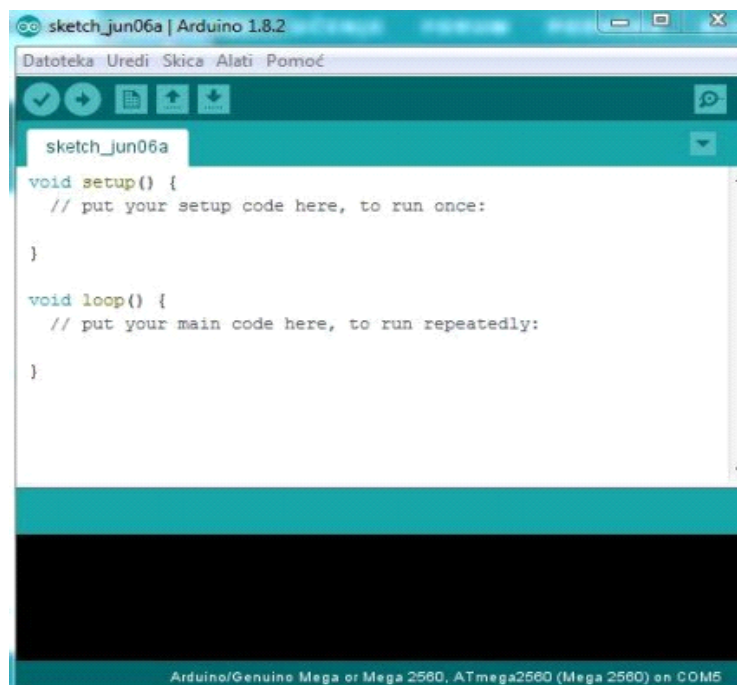
## 1.1. Zadatak preddiplomskog rada

Pomoću Arduino okruženja potrebno je napraviti maketu za sortiranje slatkiša. Na ulazu su bomboni raznih boja, a na izlazu su kutijice u kojima su bomboni sortirani prema bojama.

## 2. PRIMJENJENE TEHNOLOGIJE I ALATI

### 2.1. Arduino razvojno okruženje

Razvojno okruženje Arduino se sastoji od nekoliko gumba s funkcijama, izbornika, prostora za pisanje programskog koda te prostora za poruke, što možemo vidjeti na slici 2.1.. Uz njegovu pomoć programski se kod prenosi na Arduino uređaj, te se program napisan u Arduino razvojnom okruženju naziva „sketch“, a datoteka programa ima nastavak „.ino“. Prilikom provjere ili spremanja programa, u prostoru za poruke prikazuju se različite informacije i greške.



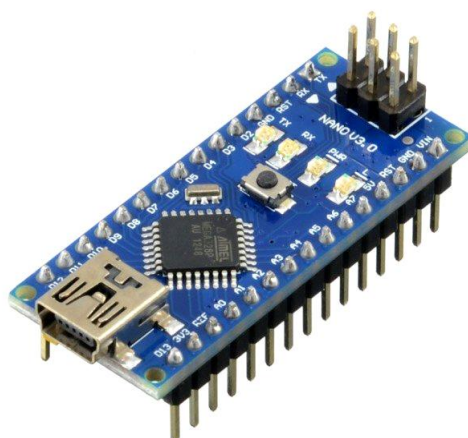
Sl. 2.1. – na slici je prikazan izgled Arduino razvojnog okruženja.

## 2.2. Arduino Nano razvojni sustav

Arduino Nano je mikro upravljačka ploča. Najbitniji dio Arduino sklopa je mikročip koji njime upravlja, a to je ATmega328p. Sklop Arduino Nano sadrži 54 ulazno/izlazna digitalna pina te 20 ulaznih analognih pinova. Frekvencija pri kojoj radi je 16 MHz. Također sadrži USB konektor, konektor napajanja, tipkalo za ponovno pokretanje te ICSP konektor koji predstavlja jedan od načina spajanja pri programiranju Arduina. Napajanje može biti preko USB konektora ili preko konektora za vanjsko napajanje na koji se može spojiti baterija ili AC/DC ispravljač. Napon vanjskog napajanje može biti od 6 V do 20 V, ali zbog mogućeg nastanka problema preporučeni napon je od 5 V do 16 V. [2][8]

*Tab. 2.1- Specifikacije Arduina Nano*

Mikrokontroler	Nano
Operativni napon	5 V
Preporučeni ulazni napon	6 V – 20 V
Granicni ulazni napon	5 V – 16 V
Digitalni U/I pinovi	54
Analogni ulazni pinovi	20
DC struja U/I pinovi	20mA
DC struja na 3.3V pinu	50mA
Flash memorija	256 KB
SRAM	8 KB
EEPROM	4 KB
Takt procesora	16 MHz



*Sl 2.2. – Vizualni prikaz Arduina Nano*

Arduino Nano sadrži osigurač koji štiti USB utor računala od kratkog spoja ili prevelike struje, te je prikazan na slici 2.2..U slučaju da se kroz USB utor isporuči više od 500 mA, osigurač prekida spoj sve dok kratki spoj nije uklonjen.Na slici 2.2 prikazan je Arduino Nano. Dimenzije su mu 43.18 mm × 18.54 mm. [4]

### **2.3. Mikro servo motor SG90**

Mikro servo motor je mal, lagan i s velikom izlaznom snagom, njegov prikaz možemo vidjeti na slici 2.3.. Ima sposobnost da rotira 180°( 90° u jednom smjeru, 90° u drugom ), te radi kao i standardni tipovi samo je manji. Za upravljanje motorom može se koristiti bilo koji servo kod, hardver ili datoteka. Dolazi s 3 roga i hardverom. [3]

*Tab. 2.2. - specifikacije servo motora SG90*

Specifikacije	Servo motor
Operativni napon	4.8 V
Operativna brzina	0.1 s/60 stupnjeva
Zakretni moment	1.8 kgf·cm
Širina mrtvog pojasa	10 μs
Raspon temperature	0 °C - 55 °C



*Sl. 2.3. - Prikaz mikro servo motora*

## **2.4. TCS3200 senzor za prepoznavanje boje**

Senzor za prepoznavanje boja TCS230 TCS3200 je kompletan detektor boja, koji se sastoji od čipova TAOS TCS3200 RGB senzora i 4 bijele led diode. Može otkriti i mjeriti gotovo neograničen raspon vidljivih boja do određenog stupnja.

Boja objekta koju možemo vidjeti zapravo je kromatsko svjetlo koje objekt reflektira u bijeloj svjetlosti (sunčeva svjetlost) nakon što apsorbira one ostale. Bijela boja je mješavina raznih vidljivih boja, što znači da ona uključuje svaku obojenu svjetlost poput crvene (R), žute (Y), zelene (G), plave (B) i ljubičasta (P). Temeljem teorije tri osnovne boje, svaka boja se proizvodi miješanjem tri osnovne boje (crvena, zelena i plava) u određenom omjeru.

Dakle, znajući udio možete dobiti boju ispitanog objekta. Za TCS3200, kada je odabran filter boja, dopušta samo određenu primarnu boju da prođe i blokira ostale dvije boje. Uz vrijednost intenziteta svjetla od tri osnovne boje, analizom možemo znati boju svjetla koja se reflektira na TCS3200. [1]

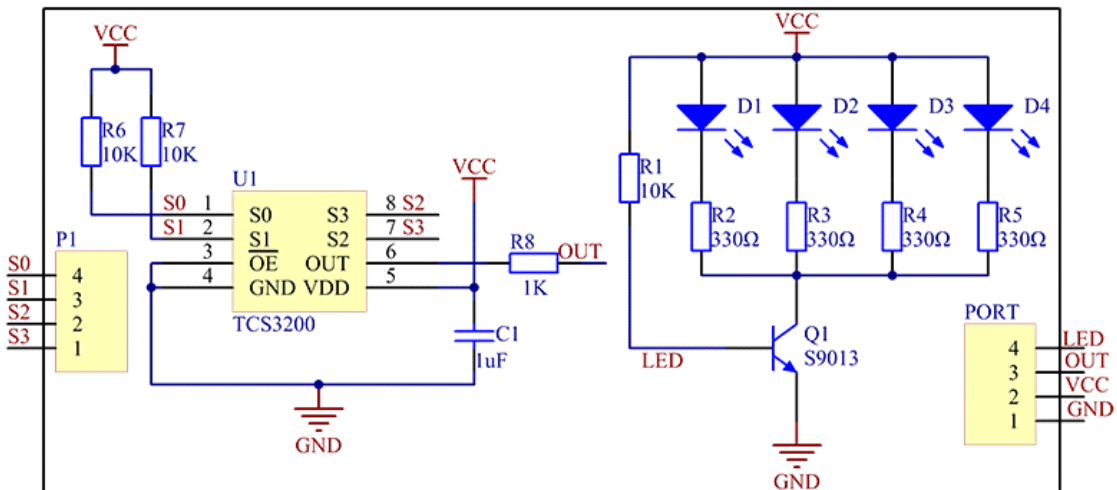


TCS3200 senzor ima 4 vrste filtera: crveni filter, zeleni filter, plavi filter i bistar bez filtera. Filter se odabire na temelju visokog / niskog stupnja S2 i S3 na modulu (pogledajte donju tablicu):

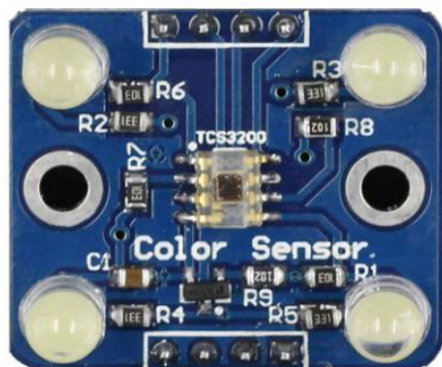
Tab 2.3. - prikaz boja filtera boja diode

S2	S3	Tip foto diode
L	L	Crvena
L	H	Plava
H	L	Prazno ( bez filtera)
H	H	Zelena

Modul sadrži programirani pretvarač koji pretvara svjetlost boje na frekvenciju. Kada dio RGB boja u svjetlu koje reflektira objekt prolazi kroz filter odabran TAOS TCS3200 RGB čipu, ugrađeni oscilator emitira kvadratne valove, čiji prikaz vidimo na slici 2.5. i 2.6.. Frekvencija valova izravno je proporcionalna intenzitetu svjetla - što je intenzivnije svjetlo, to je veća učestalost. Također, frekvencija vanjske utičnice na modulu senzora je proporcionalna onoj oscilatora; udio ovisi o visokom / niskom stupcu S0 i S1, kao što je prikazano u tablici. [9]



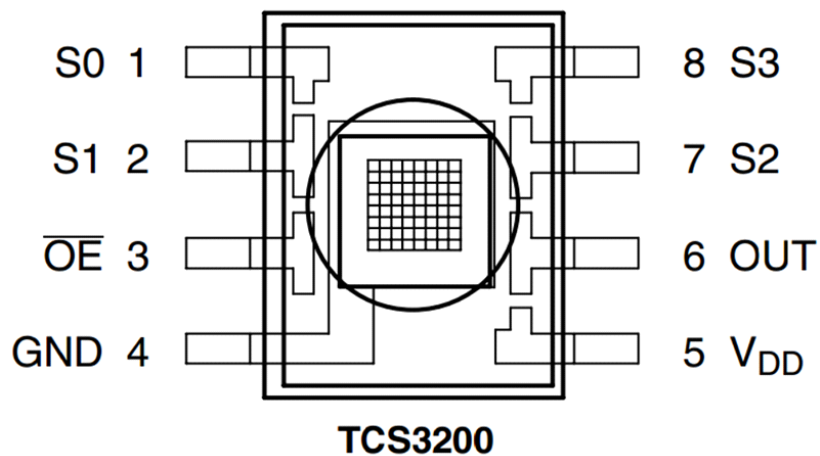
Sl. 2.4. - Shematski dijagram modula



Sl. 2.5. - srednja strana senzora



2.6. - stražnja strana senzora



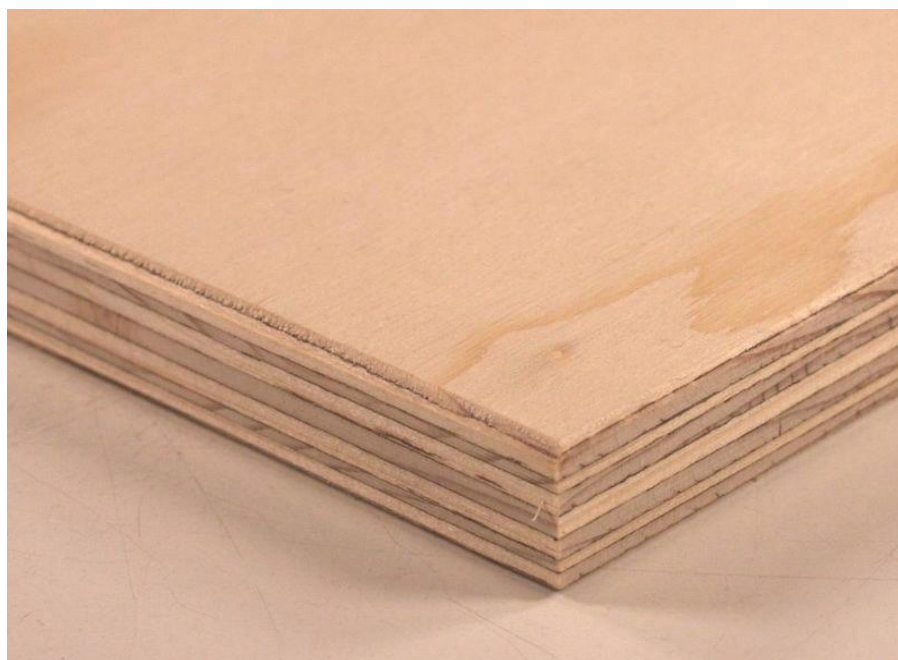
Sl. 2.7. - pin funkcije TCS3200 senzora

Tab 2.4. - legenda ulaza senzora

PIN	Ime	I/O	Opis
4	GND		Uzemljenje napajanja. Svi naponi su uzemljeni u GND.
3	OE	I	Omogući za fo
6	OUT	O	Izlazna frekvencija (fo).
1, 2	S0, S1	I	Izlazi za skaliranje izlazne frekvencije.
7, 8	S2, S3	I	Ulazi za odabir tipa fotodioda.
5	VDD		VDD

## 2.5. Šperploča 4mm

Šperploča je prešano drvo. Koristi se za izolaciju i izradu maketa. Također se koristi za stvaranje različitih modela kuća i gradova. Ona je lagana, meka, čvrsta i ne porozna. Šperploča se dobiva rezanjem i lijepljenjem neparnog broja slojeva furnira, koji se slaže jedan iznad drugog križa. To povećava snagu šperploče kao materijala, njen prikaz možemo vidjeti na slici 2.8.. Koristi se u obrtu i proizvodnji namještaja, stolarije i ostalih predmeta. [5]



Sl. 2.8. – prikaz šperploče od 4mm koju smo koristili za izradu makete

## 2.6. Laserski rezač CO<sub>2</sub>

Rezanje laserom je tehnologija koja koristi laser za rezanje materijala, te se uglavnom koristi za industrijsku primjenu, ali sve češće se koristi i u školama, malim poduzećima i kod hobija. Rezanje laserom radi tako da se direktno kontrolira izlazna snaga lasera, djelujući obično kompjuterom. Materijal se ili topi, ili izgara, ili isparava, ili ga otpuhne mlaz zraka pod pritiskom, ostavljajući visoko kvalitetne rubove reza. Ne režu se samo limovi, nego i profili i cijevi. Tehnika koju sam koristio pri ovom radu je bila tehnika izgaranja.

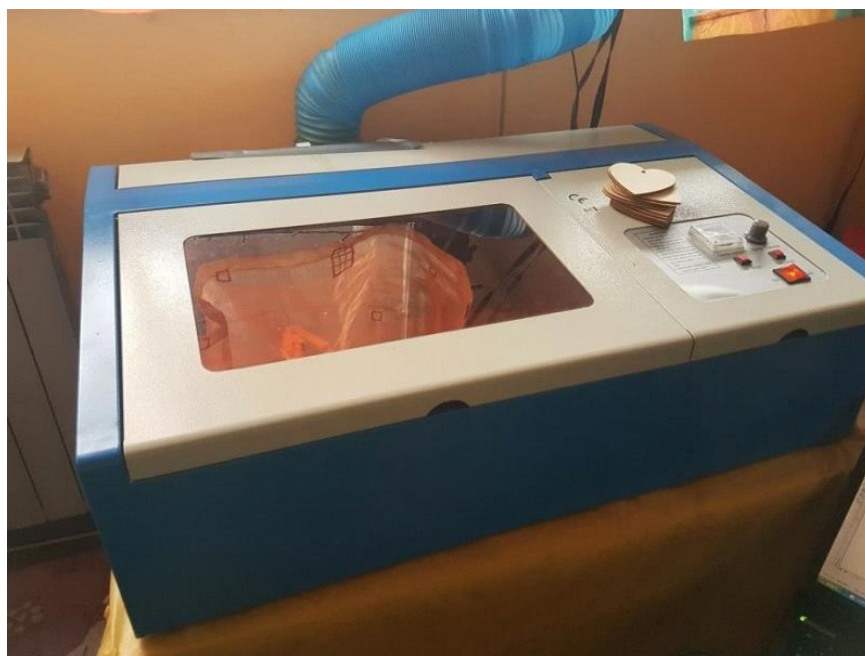
Tab 2.5. - prikaz vrsta rezača

Laserski medij	Primjena
CO <sub>2</sub>	Bušenje Rezanje/označavanje, graviranje
Nd	Visokoenergetskiimpulsi Mala brzina ponavljanja (1 kHz) Bušenje
Nd-YAG	Vrlovisokienergetskiimpulsi Bušenje, graviranje, podešavanje

Kod rezanja laserom koriste se 3 vrste lasera. CO<sub>2</sub> laser je pogodan za rezanje, bušenje i graviranje. Nd laser i Nd:YAG laseri su slični laseri, a razlikuju se po primjeni. Nd laser se koristi za bušenje, gdje je potreba velika snaga, ali s malo ponavljanja. Nd:YAG laser se koristi za velike snage kod bušenja i graviranja, s velikim brojem ponavljanja. Svi ovi laseri se mogu koristiti i kod zavarivanja.

## 2.7. Postupak rezanja

Za rezanje sam koristio rezanje plazmom, što je obično debljina laserske zrake manja od 0,3 mm, a moguće je ostvariti debljinu rezanja i manju od 0,1 mm. Kod rezanja treba napomenuti da prvo treba izbušiti rupu kroz materijal, a za to se koristi velika snaga lasera, i traje obično 5 do 15 sekundi, prikaz laserskog rezača možemo pogledati na slici 2.9.. [6] [7]



*Sl. 2.9. - Laserski rezač korišten za rezanje šper ploča*

## 2.8. Testiranje sustava

Sustav je testiran tako što smo u stalak za bombone ubacili 12 bombona različitih boja (plavi, crveni, zeleni i žuti). Kako bi provjerili ispravnost cijelog sustava bilo je potrebno da sustav sortira svaki bombon u predviđenu posudu za određenu boju bombona. Tok prolaska bombona kroz maketu nije prošao po planu, zbog senzora za boju. Najčešće miješane boje su bile narančasta i crvena. Vanjska svjetlost okoline je također stvarala problem. Prvo smo testiranje započeli s plavim bombonima, te je u prvom slučaju prolaznost bila 9/12. Zatim smo testirali crvene bombone, čija je prolaznost bila najniža, 6/12. U trećem i četvrtom slučaju sa zelenim i žutim bombonima smo imali manji problem jer je prolaznost bila 7/12 za zelene bombone i 10/12 za žute bombone.

Tab. 2.6. – prikaz rezultata sortiranja bombona

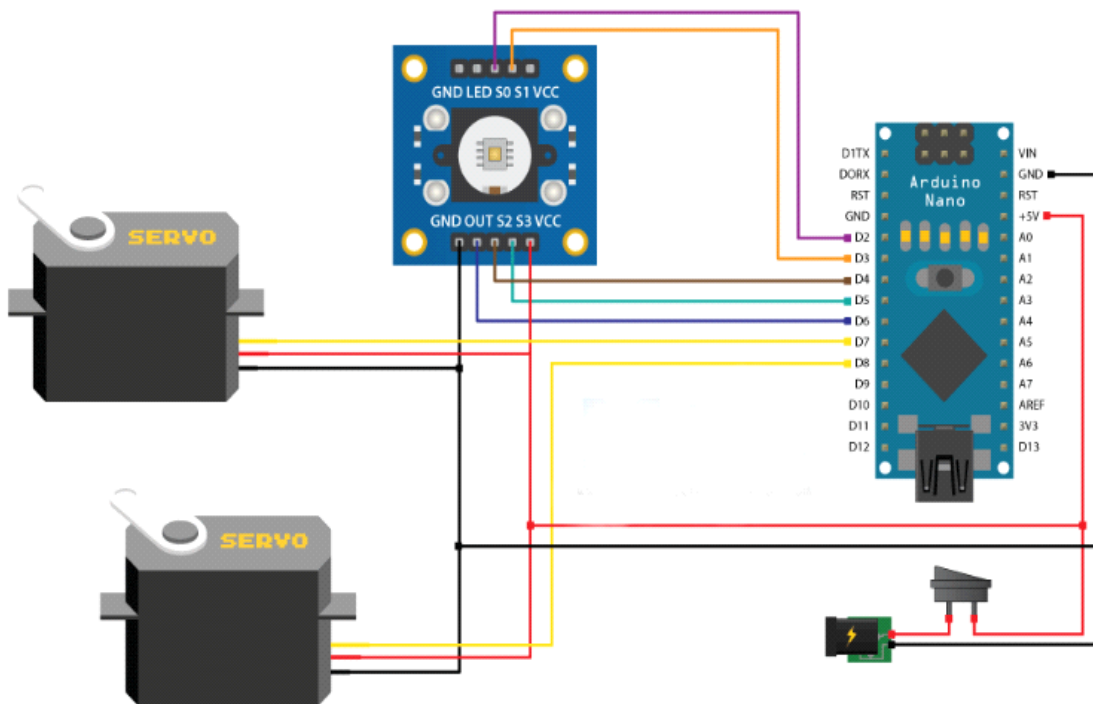
Sorter boja	Datum	Test	Testni slučaj	Broj točnih sortiranja (od 12)	Broj pogrešnih sortiranja (od 12)
Plava	23.8.2018.	Sortiranje	Provjeri hoće li bombon biti sortiran u predviđenu posudu	9	3
Crvena	23.8.2018.	Sortiranje	Provjeri hoće li bombon biti sortiran u predviđenu posudu	6	6
Zelena	23.8.2018.	Sortiranje	Provjeri hoće li bombon biti sortiran u predviđenu posudu	7	5
Žuta	23.8.2018.	Sortiranje	Provjeri hoće li bombon biti sortiran u predviđenu posudu	10	2

### 3. REALIZACIJA SUSTAVA

Kao što je već objašnjeno u uvodu, potrebno je izraditi sorter bombona. Potrebno je bilo prvo narezati potrebne materijale pravilnih dimenzija, spojiti ih, te ukomponirati s elektronikom koju smo naveli. Nakon toga je bilo potrebno isprogramirati sklop kako bi obavljao zadane funkcije.

#### 3.1. Elektronička shema sustava

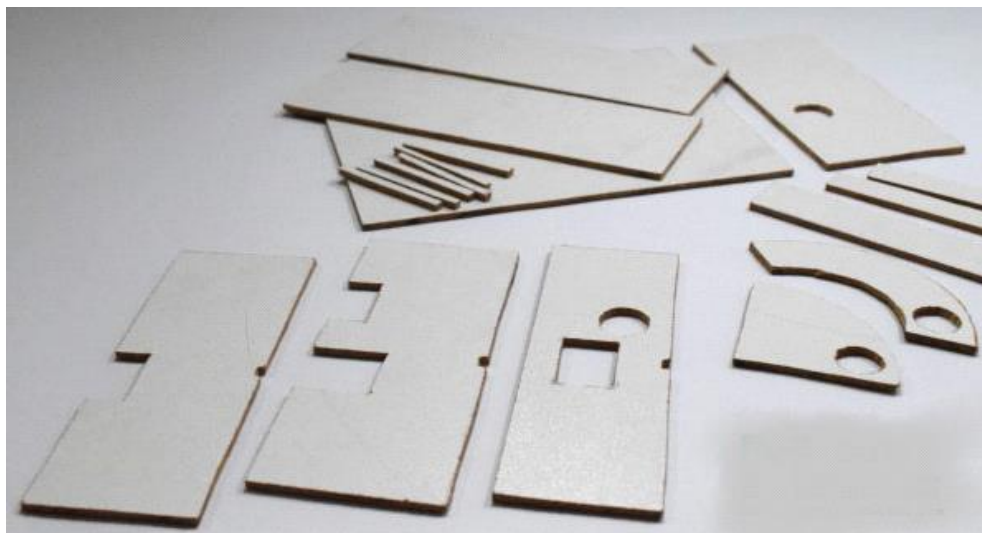
Sustav smo spojili po elektroničkoj shemi sustava koja je prikazana na slici ispod.



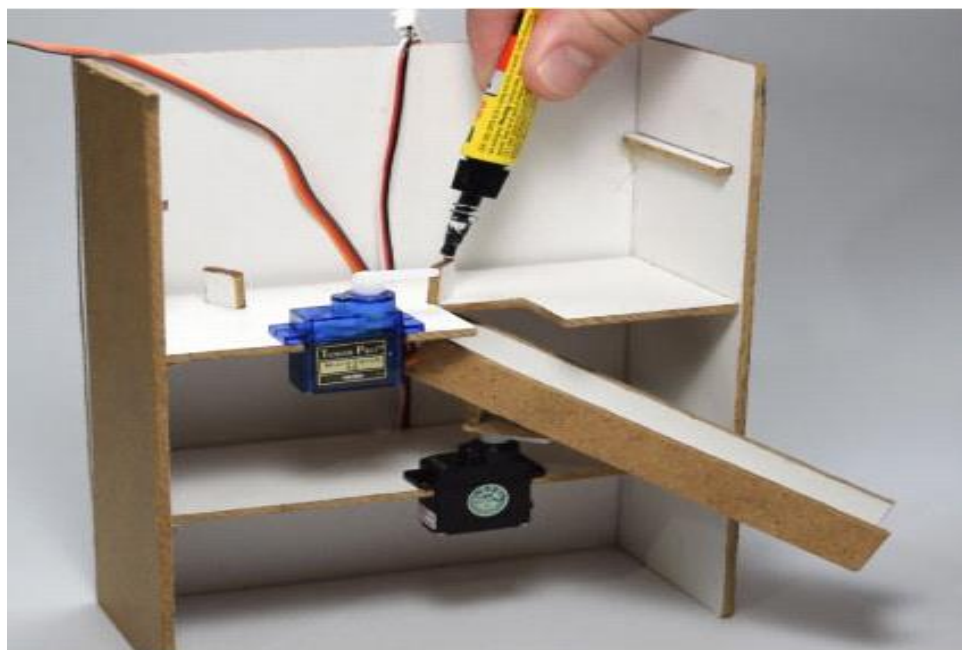
Sl. 3.1. – prikaz sheme komponenti sustava

#### 3.2. Izrada makete sustava

Prvo smo sve plohe i bridove izmjerili, te izrezali pomoću laserskog rezača, što možemo vidjeti na slici 3.2., te smo nakon toga sve plohe polijepili pomoću brzog ljepila i ljepila za drvo (slika 3.3.). U idućem koraku smo sve elektronske uređaje spojili pomoću vodiča (Sl. 3.4.)

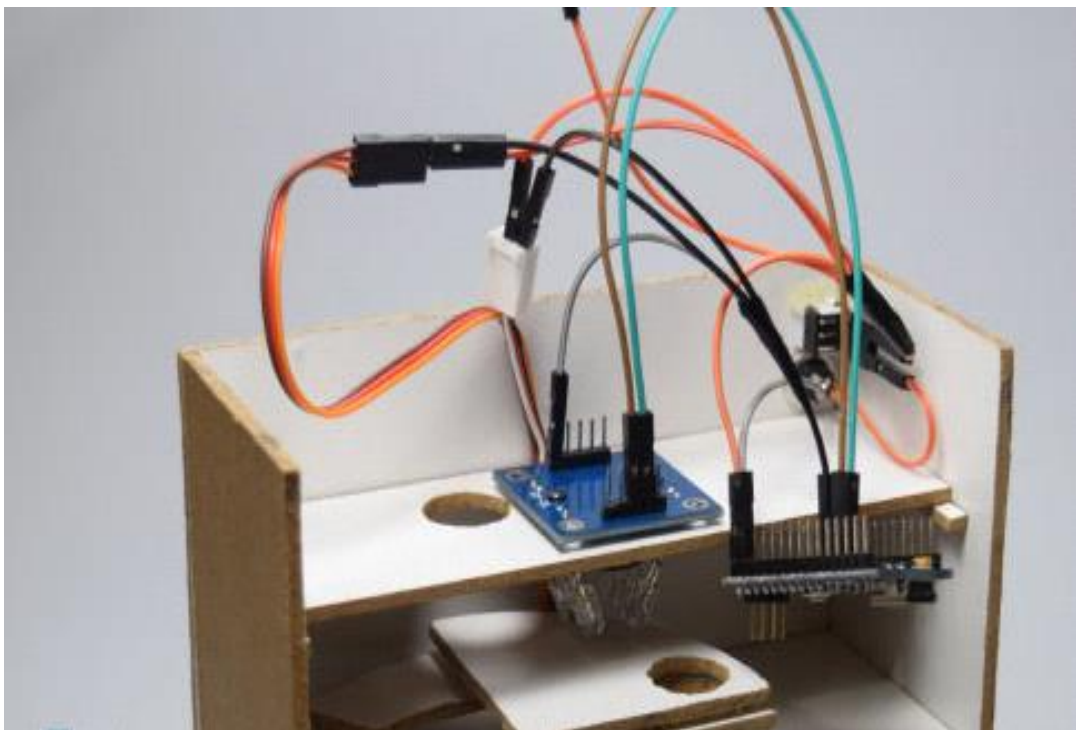


*Sl 3.2. - prikaz po mjeri odrezanih konstrukcijskih dijelova*



*Sl 3.3. - prikaz dovršetka ljepljenja i sastavljanje el. sklopova*





Sl 3.4. - prikaz spojenih sklopova na konstrukciji

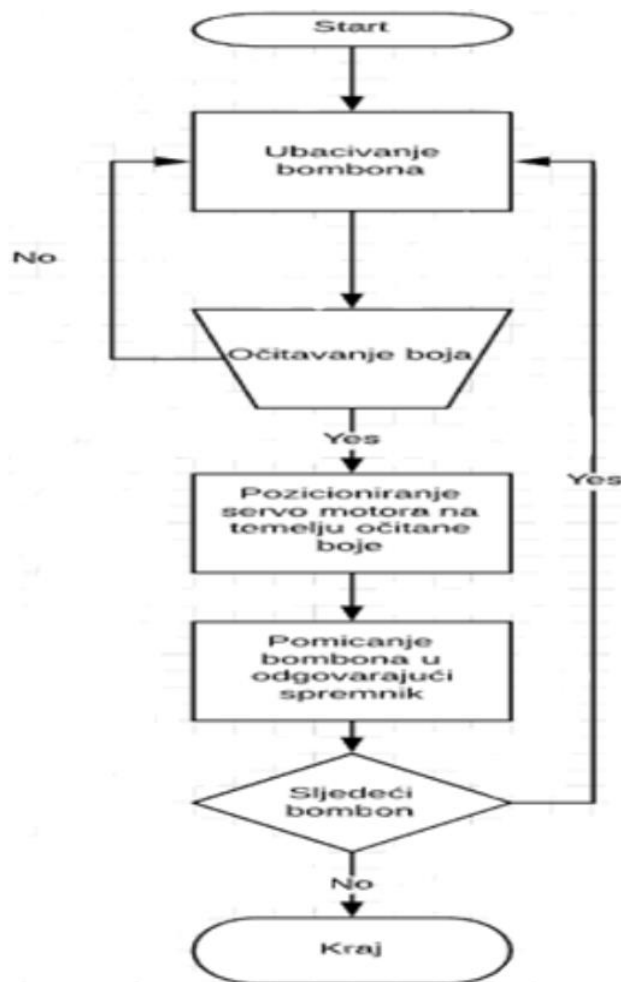
### 3.3. Programiranje sklopovlja

Prilikom programiranja softverskog dijela koristio sam Arduino programming language. Pri početku pisanja koda smo morali uključiti biblioteku "Servo.h", odrediti igle na koje će se spojiti sensor boje, stvoriti servo objekte i iznijeti neke varijable potrebne za program. U sekciji za postavljanje potrebno je definirati igle kao izlaze i ulaze, postaviti skaliranje frekvencije za senzor u boji, definirati servo igle i pokrenuti serijsku komunikaciju za ispis rezultata čitane boje na serijskom monitoru.

U sekciji petlje, naš program počinje pomicanjem gornjeg servo motora na položaj akumulatorskog punjača.

Sljedeće pomoću *for* petlje rotirat ćemo i staviti kuglu na položaj senzora boje. Koristimo *for* petlju, tako da možemo kontrolirati brzinu rotacije promjenom vremena kašnjenja u petlji.

Zatim, nakon pola sekunde kašnjenja, pomoću funkcije custom made, readColor () čitat ćemo boju elementa kojeg sortiramo. Senzor čita 3 različite vrijednosti za svaku kuglicu, crvenu, zelenu i plavu, a prema tim vrijednostima kažemo koja je stvarna boja. Programski kod sortera boja možemo naći u prilogima. Očitavanje boja pomoću senzora, te pomicanje servo motora na određeni položaj vidljivo je na Sl. 3.5.. Prikaz toka sustava za raspoređivanje bombona prikazan je blok dijagramom 3.1..



*Blok dijagram 3.1. – prikaz toka sustava*

```
1  color = readColor();
2  delay(10);
3
4  switch(color){
5  case1:
6  bottomServo.write(50);
7  break;
8
9  case2:
10 bottomServo.write(75);
11 break;
12
13 case3:
14 bottomServo.write(100);
15 break;
16
17 case4:
18 bottomServo.write(125);
19 break;
20
21 case5:
22 bottomServo.write(150);
23 break;
24
25 case6:
26 bottomServo.write(175);
27 break;
28
29 case0:
30 break;
31 }
```

*Sl. 3.5. prikaz koda za određivanje spremnika bombona*

## **4. ZAKLJUČAK**

Ovim završnim radom postignuta je funkcionalna maketa za sortiranje bombona po bojama. Konstrukcijskim dijelom steknuta su poznanstva o preciznom rezanju, mjerenju i sastavljanju maketa, a u softverskom dijelu stečeno je novo znanje u radu Arduino programskom jeziku. Uspješno je dokazano da se jednostavno u kućnoj izradi može napraviti automatski programiran sklop koji će obavljati radnju koja mu je zadana. Stečeno je znanje koje bi se uz dodatno nadograđivanje moglo koristiti pri korištenju većih sustava i programa koji se mogu koristiti u masovnoj proizvodnji za izradu i sortiranje određenih elemenata.

## LITERATURA:

- [1] Wikipedia, Color sensor module,  
[http://wiki.sunfounder.cc/index.php?title=Color\\_Sensor\\_Module](http://wiki.sunfounder.cc/index.php?title=Color_Sensor_Module)
- [2] Wikipedia, Arduino, <https://en.wikipedia.org/wiki/Arduino>
- [3] Wikipedia, Servo Motor, <https://en.wikipedia.org/wiki/Servomotor>
- [4] Wikipedia, Nano, [http://www.geeetech.com/wiki/index.php/Arduino\\_Nano](http://www.geeetech.com/wiki/index.php/Arduino_Nano)
- [5] Wikipedia, Šper ploča,  
<https://sr.wikipedia.org/wiki/%C5%A0perplo%C4%8Da>
- [6] Wikipedia, Rezanje laserom, [https://hr.wikipedia.org/wiki/Rezanje\\_laserom](https://hr.wikipedia.org/wiki/Rezanje_laserom)
- [7] Wikipedia, Rezanje plamenom,  
[https://hr.wikipedia.org/wiki/Rezanje\\_laserom#Rezanje\\_plamenom](https://hr.wikipedia.org/wiki/Rezanje_laserom#Rezanje_plamenom)
- [8] Wikipedia, Arduino platforma,  
[https://hr.wikipedia.org/wiki/Arduino#Minimalni\\_Arduino\\_program](https://hr.wikipedia.org/wiki/Arduino#Minimalni_Arduino_program)
- [9] Wikipedia, Senzor za prepoznavanje boja, [TCS3200\\_TCS3210\\_datasheet.pdf](#)

## SAŽETAK

**Naslov:** IZRADA MAKETE UREĐAJA ZA SORTIRANJE SLATKIŠA PREMA BOJI

Ovim završnim radom postignuta je funkcionalna maketa za sortiranje bombona po bojama. Stečeno je znanje koje bi se uz dodatno nadograđivanje moglo koristiti pri korištenju većih sustava i programa koji se mogu koristiti u masovnoj proizvodnji za izradu i sortiranje određenih elemenata.

**Ključne riječi:** sorter, bomboni, Arduino, sensor za boje, transport

## **ABSTRACT**

**Title:** MAKING OF MODEL FOR SORTING CANDIES BY COLOR

This final work has accomplished a functional model for sorting candies by colors. Knowledge has been gained that could be used with additional upgrades when using larger systems and programs that can be used in a mass production to produce and sort certain elements.

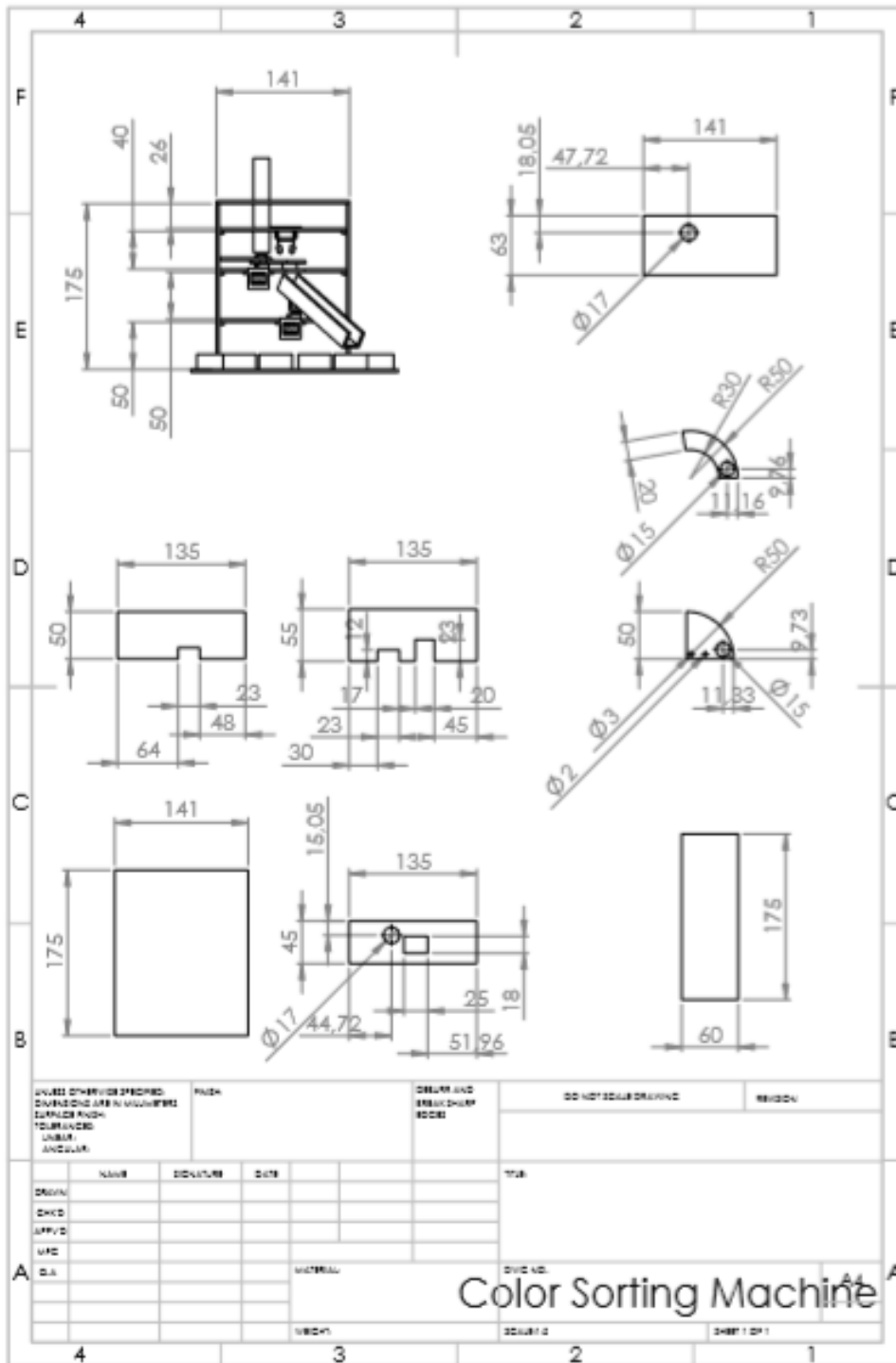
**Keywords:** sorter, candy, Arduino, color sensor, transport

## **ŽIVOTOPIS**

Luka Neseck rođen je 16. ožujka 1996. u Osijeku. U Osijeku završava osnovnu školu „Frana Krste Frankopana“, te 2011. Upisuje Jezičnu gimnaziju u Osijeku. Godine 2015. Ostvaruje upis na Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer računarstvo.

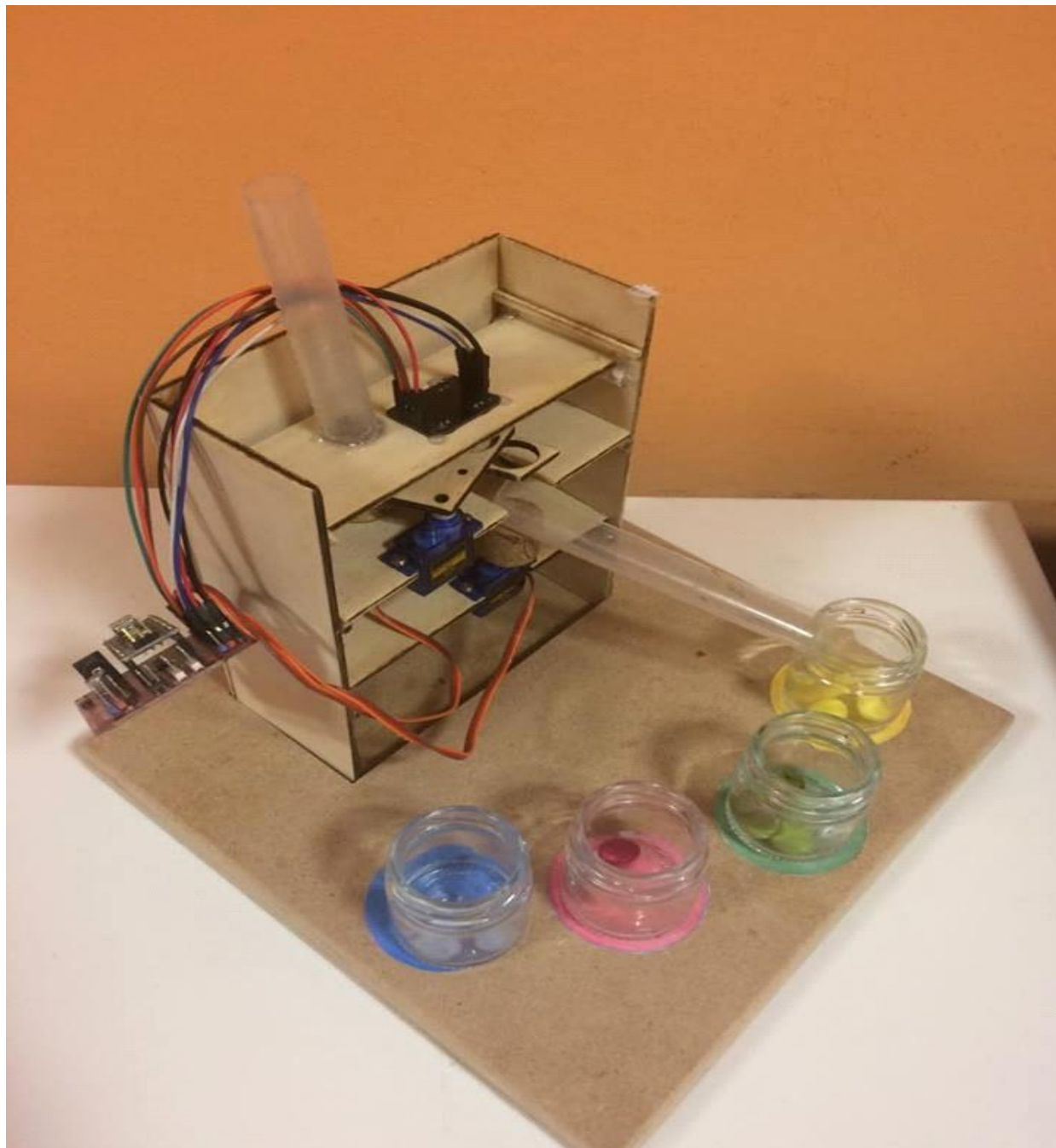


**PRILOG 1) Kompletna shema nacрта konstrukcije sustava**

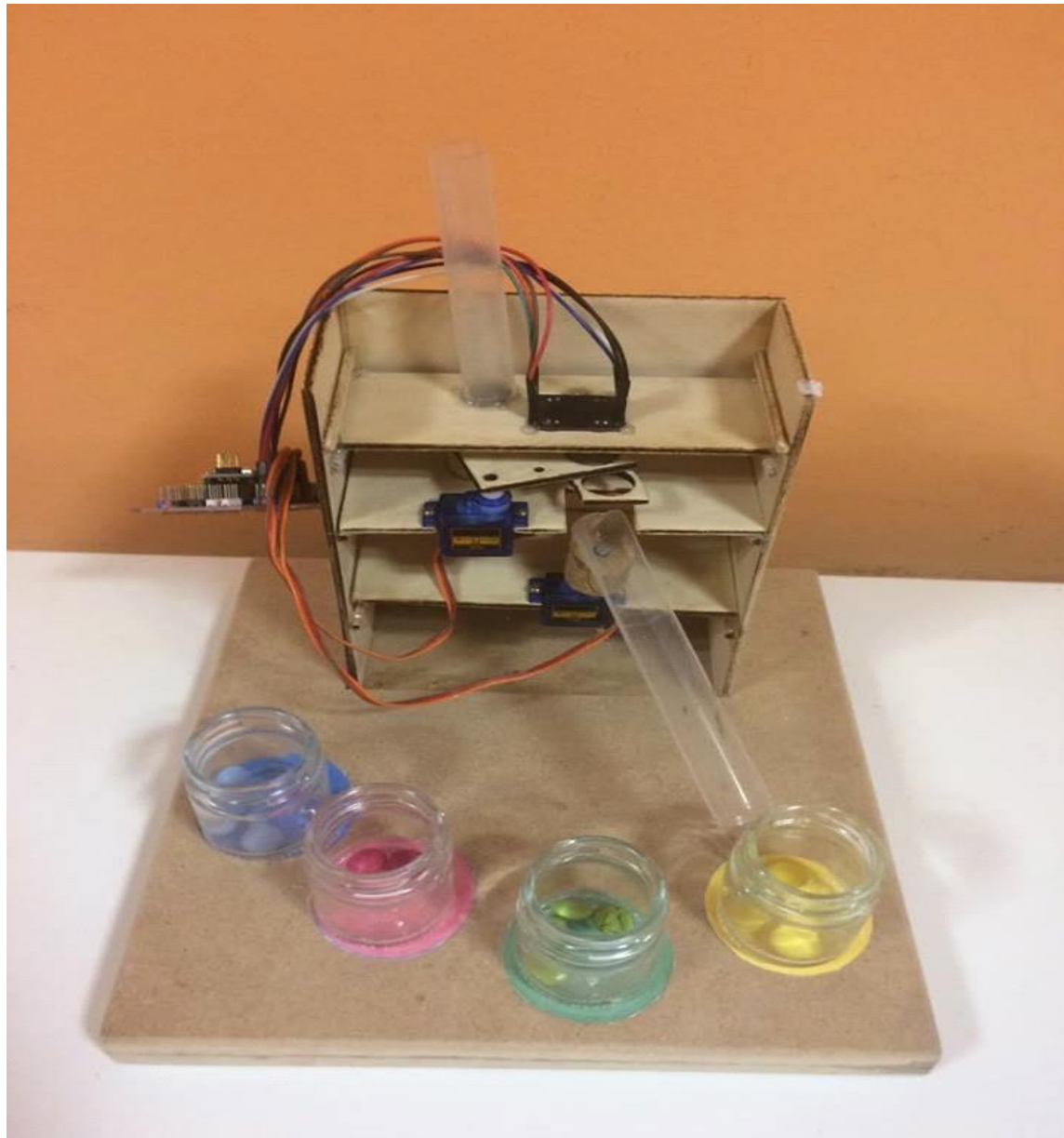


Sl. 4. – Shema sustav

**PRILOG 2) Slika sustava u pogonu**



*Sl. 5.1. – prikaz sustava*



Sl. 5.2. – prikaz sustava

### PRILOG 3) Program za Arduino razvojni sustav

```
#include <Servo.h>

#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6

Servo topServo;
Servo bottomServo;

int frequency = 0;
int color=0;

voidsetup(){
pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensorOut, INPUT);

digitalWrite(S0, HIGH);
digitalWrite(S1, LOW);

topServo.attach(7);
bottomServo.attach(8);

Serial.begin(9600);
}

voidloop(){

topServo.write(115);
delay(500);

for(int i = 115; i >65; i--){
topServo.write(i);
delay(2);
}
delay(500);

color = readColor();
```

```
delay(10);

switch(color){
case1:
bottomServo.write(50);
break;

case2:
bottomServo.write(75);
break;

case3:
bottomServo.write(100);
break;

case4:
bottomServo.write(125);
break;

case5:
bottomServo.write(150);
break;

case6:
bottomServo.write(175);
break;

case0:
break;
}
delay(300);

for(int i = 65; i >29; i--){
topServo.write(i);
delay(2);
}
delay(200);

for(int i = 29; i <115; i++){
topServo.write(i);
delay(2);
}
color=0;
}
```

```

intreadColor(){
digitalWrite(S2, LOW);
digitalWrite(S3, LOW);

frequency = pulseIn(sensorOut, LOW);
int R = frequency;

Serial.print("R= ");
Serial.print(frequency);
Serial.print(" ");
delay(50);

digitalWrite(S2, HIGH);
digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);
int G = frequency;

Serial.print("G= ");
Serial.print(frequency);
Serial.print(" ");
delay(50);

digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);
frequency = pulseIn(sensorOut, LOW);
int B = frequency;
Serial.print("B= ");
Serial.print(frequency);
Serial.println(" ");
delay(50);

if(R<45& R>32& G<65& G>55){
color = 1; // Red
}
if(G<55& G>43& B<47&B>35){
color = 2; // Orange
}
if(R<53& R>40& G<53& G>40){
color = 3; // Green
}
if(R<38& R>24& G<44& G>30){
color = 4; // Yellow
}
if(R<56& R>46& G<65& G>55){

```

```
color = 5; // Brown
}
if(G<58& G>45& B<40&B>26){
color = 6; // Blue
}
return color;

}
```