Sustav za sortiranje objekata s primjenom u poljoprivrednoj proizvodnji

Crnoja, Denis

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij

SUSTAV ZA SORTIRANJE OBJEKATA S PRIMJENOM U POLJOPRIVREDNOJ PROIZVODNJI

Završni rad

Denis Crnoja

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju

Ocjena završnog rada na stručnom prijediplomskom studiju

Ime i prezime pristupnika:	Denis Crnoja
Studij, smjer:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. pristupnika, god.	A4518, 23.07.2018.
JMBAG:	0165076462
Mentor:	prof. dr. sc. Tomislav Keser
Sumentor:	prof. dr. sc. Damir Blažević
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Tomislav Galba
Član Povjerenstva 1:	prof. dr. sc. Tomislav Keser
Član Povjerenstva 2:	izv. prof. dr. sc. Alfonzo Baumgartner
Naslov završnog rada:	Sustav za sortiranje objekata s primjenom u poljoprivrednoj proizvodnji
Znanstvena grana završnog rada:	**Procesno računarstvo (zn. polje računarstvo)**
Zadatak završnog rada:	Razviti, izraditi i testirati sustav za sortiranje poljoprivrednih proizvoda po boji i masi. Sustav implementirati u sortiranju paprike, raznih veličina i ostalih fizionomskih značajki. Omogućiti sortiranje proizvoda u svrhu odvajanja i preciznog vaganja i pakiranja.

Datum ocjene pismenog dijela završnog rada od strane mentora:	23.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Dobar (3)
Datum obrane završnog rada:	4.10.2024
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Vrlo dobar (4)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	08.10.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 08.10.2024.

Ime i prezime Pristupnika:	Denis Crnoja
Studij:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. Pristupnika, godina upisa:	A4518, 23.07.2018.
Turnitin podudaranje [%]:	13

Ovom izjavom izjavljujem da je rad pod nazivom**: Sustav za sortiranje objekata s primjenom u poljoprivrednoj proizvodnji**

izrađen pod vodstvom mentora prof. dr. sc. Tomislav Keser

i sumentora prof. dr. sc. Damir Blažević

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD

Sustavi za sortiranje objekata u poljoprivrednoj proizvodnji imaju ključnu ulogu u povećanju efikasnosti, kvalitete proizvoda i profitabilnosti. Ovi sustavi omogućuju automatsko sortiranje poljoprivrednih proizvoda na temelju različitih kriterija poput oblika, boje ili mase. Postojeća rješenja za sortiranje objekata uključuju korištenje mehaničkih, optičkih i elektroničkih tehnologija. Svaka ima svoje prednosti i nedostatke. Mehanički sustavi za sortiranje su među najstarijim metodama i uključuju korištenje različitih fizičkih barijera i mehaničkih pomagala za razdvajanje objekata. Ovi sustavi se koriste vibracijskim rešetkama ili transportnim trakama s otvorima različitih dimenzija. Iako su relativno jednostavni i jeftini dosta su ograničeni jer su skloni trošenju i kvarovima. Optički sustavi koriste senzore i kamere za detekciju vanjskih karakteristika proizvoda. Izuzetno su precizni u prepoznavanju karakteristika poput boje ili površinskih oštećenja. Ovakvi sustavi su široko rasprostranjeni u industriji voća i povrća zbog njihove preciznosti i velike brzine rada. Elektronički sustavi se koriste raznim senzorima poput ultrazvučnih, infracrvenih za analizu unutarnjih karakteristika proizvoda. Ovi sustavi mogu detektirati unutarnje mane proizvoda ili moguće razlike u gustoći. Njihova primjena je najraširenija u industrijama orašastih plodova i drugih plodova gdje nam je unutarnja kvaliteta proizvoda bitna. Motiv za odabir ovoga rada proizlazi iz potrebe povećanja efikasnosti i smanjenja vremena potrebnog za sortiranje proizvoda uz smanjenje troškova i gubitaka. Cilj je napraviti sustav za sortiranje kombinacijom mehaničkih i elektroničkih tehnologija. Na ovaj način želimo ostvariti bolju kontrolu kvalitete i smanjenje otpada proizvoda te ubrzati proces sortiranja. Sustav će biti dovoljno fleksibilan kako bi se mogao prilagoditi različitim vrstama proizvoda i zahtjevima.

1.1. Zadatak i struktura zadatka

Zadatak ovoga rada je napraviti sustav za sortiranje proizvoda po boji i masi koji kombinira razne mehaničke i elektroničke tehnologije. Sustav bi trebao biti sposoban prepoznati i sortirati proizvode na osnovu vanjskih karakteristika, u ovome slučaju boje. Razvijeni sustav je prilagodljiv različitim vrstama proizvoda i jednostavan za implementaciju i održavanje, time omogućujemo povećanje učinkovitosti u poljoprivrednoj proizvodnji. Cilj je stvoriti sustav na precizan i ekonomičan način koji će pomoći proizvođačima da povećaju produktivnost i kvalitetu proizvoda te smanje otpad. Na kraju rada očekuje se dobivanje prototipa sustava koji će biti testiran u stvarnim uvjetima i dokumentacije koja će detaljno opisati sve korake razvoja i implementacije sustava. U sljedećem poglavlju prijedlog sklopovskog rješenja prikazujemo idejni rad sustava pomoću blokovskog dijagrama. Cilj je prikazati odnose između komponenata, točnije kako se one međusobno odnose i kako utječu jedna na drugu. Poglavlje prijedlog programskog rješenja prikazuje i opisuje blokovski dijagram algoritma za programsko rješenje sustava za sortiranje objekata po boji i masi. Dijagram prikazuje uzročno-posljedične veze između programskih segmenata i funkcije koje će se izvršavati kako bi se postiglo sortiranje objekta.

2. SUSTAV ZA SORTIRANJE POMOĆU SENZORA BOJA, VAGE I POKRETNE TRAKE

Kako bi sortirali proizvod po boji potrebno nam je očitati, tj. pretvoriti fizičku karakteristiku proizvoda, boju u neku vrijednost, varijablu. Tu varijablu je potrebno poslati preko mikroračunala do programskog koda u kojemu obrađuje taj ulazni signal i ovisno o njegovom iznosu određuje se daljnji rad sustava.

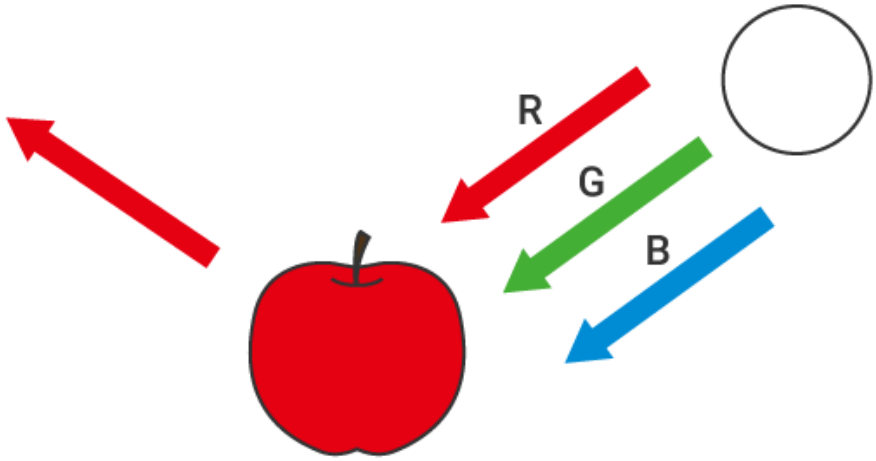
Rad sustava određuje i masa proizvoda. Kako bi se očitala napunjenost spremnika u kojeg idu proizvodi, potrebno je znati točno kolika je ukupna masa proizvoda unutar spremnika. Ako je spremnik pun treba zaustaviti program, prikazati radniku za pogonom da je pun i isprazniti ga. Spremnici u poljoprivredi su veliki, stoga je potrebna velika vaga za spremnike ili je potrebno izvagati svaki proizvod koji ulazi u spremnik, kao što ćemo mi u ovome sustavu napraviti. Važe se svaki ulazni proizvod i u programskome kodu se zbraja svaki i sprema u varijablu spremnika određene boje.

Kako u sustavu moramo koristiti senzor za boju i vagu, trebamo nešto što pomiče plodove sa vage nakon što su očitane vrijednosti mase. Za to bi se koristio servo motor sa nekakvom pregradom koja gura proizvode. Za prijenos proizvoda prema spremnicima potrebno nam je kreirati pokretnu traku. Nakon što motor pogura proizvod sa vage potrebno nam je postaviti pokretnu traku ispod vage tako da se proizvod nastavi kretati prema spremnicima. Kako bi svaki proizvod našao put do svoga spremnika iznad trake moramo iskoristiti još servo motora sa pregradom koja bi se postavila preko pokretne trake te uputila proizvod sa trake i smjestila ga u određeni spremnik.

U maketi nam je potrebno vodičima povezati sve te hardvere: senzore, motore, zaslone. Za to povezivanje rada svih hardverskih jedinica nam je potrebno koristiti nekakvo mikroračunalo koje prenosi signale sa i u programski kod koji kontrolira rad sustava. Sve te hardverske jedinice je potrebno povezati i fizički napraviti maketu kako bi obavljale svoju zadaću.

2.1. Teorijski osvrt na očitanje boje i određivanje mase

Bijela svjetlost sastavljena je od kontinuiranog niza svih boja vidljivog spektra. Bojom nekog tijela smatramo boju koje tijelo reflektira kada je osvijetljeno bijelom svjetlošću,. tj tijelo je obojeno bojom ako površina upije bijelu svjetlost, ali samo na određenom valnom području. Tvar koja tako upije bijelu svjetlost, a reflektira crveno nama je crvena boja. Npr. slika 2.1. jabuka je crvena zato što se sve boje osim crvene upijaju unutar jabuke, a samo se ta crvena boja reflektira. U sustavu mi kreiramo umjetni izvor bijelog svjetla usmjerenog prema proizvodu, tako stvorimo odbijanje tog svijetla od proizvod. Odbijenu svjetlost možemo očitati i obraditi pomoću hardvera kako bi dobili varijablu za boju proizvoda u programskome kodu.

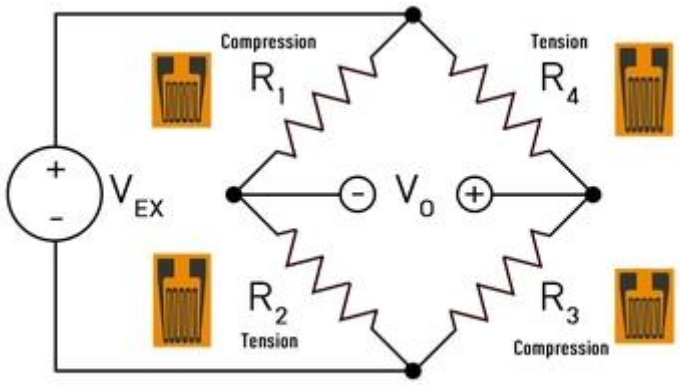


Slika 2.1. *Reflektiranje boje (R-crvena, G-zelena, B-plava boja)*[1]

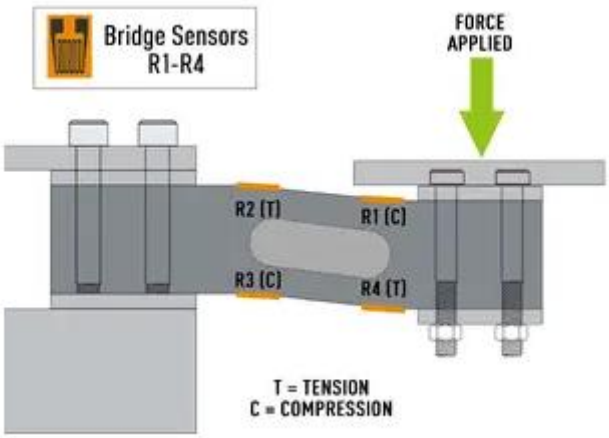
U realizaciji našega rješenja mjerenja mase smo koristili zakone sile teže. Sila teža je rezultanta privlačne sile Zemlje i centrifugalne sile zbog Zemljine rotacije, a gravitacija je sila uzajamnog privlačenja između masa. Po Isaacu Newtonu, „opći zakon gravitacije“, gravitacija je osnovno svojstvo mase. Masu mjerimo pomoću sile naprezanja, uz otpore i napone Wheatstoneovog mosta unutar mjerne ćelije. Wheatstoneov most je mjerni most za mjerenje električnoga otpora. On se sastoji od četiri otpornika spojena u četverokut, slika 2.2. Vex napon na slici je poznati napon izvora mosta, Vo napon se mjeri. Kada je omjer prvog drugog i trećeg i četvrtog otpornika jednak onda je napon Vo nula. Ako postoji promjena u vrijednosti nekog otpornika, Vo će imati promjenu koja se može izmjeriti i protumačiti pomoću Ohmovog zakona. Ohmov zakon kaže da je struja (I) koja teče kroz vodič između dvije točke izravno proporcionalna naponu (U) na dvije točke. Uvodi se otpor (R) kao konstanta u ovom odnosu, neovisan je o struji. Ohmov zakon se izražava jednadžbom: I= U/R.

U mjernoj ćeliji, slika 2.3. se otpornici Wheatstoneovog mosta zamjenjuju mjeračima naprezanja u izmjeničnim mjerenjima *tension* (napetosti) i *compression* (kompresije). Kada se primjeni sila, *force applied* (primijenjena sila) na slici 2.3., otpor u svakom *bridge sensoru* (mjeraču naprezanja) se mijenja i Vo se mjeri. Iz dobivenih podataka Vo se može lako odrediti pomoću jednadžbe[2]:

$$V_O = \left[\frac{R3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right] \times V_{EX} \qquad [2]$$

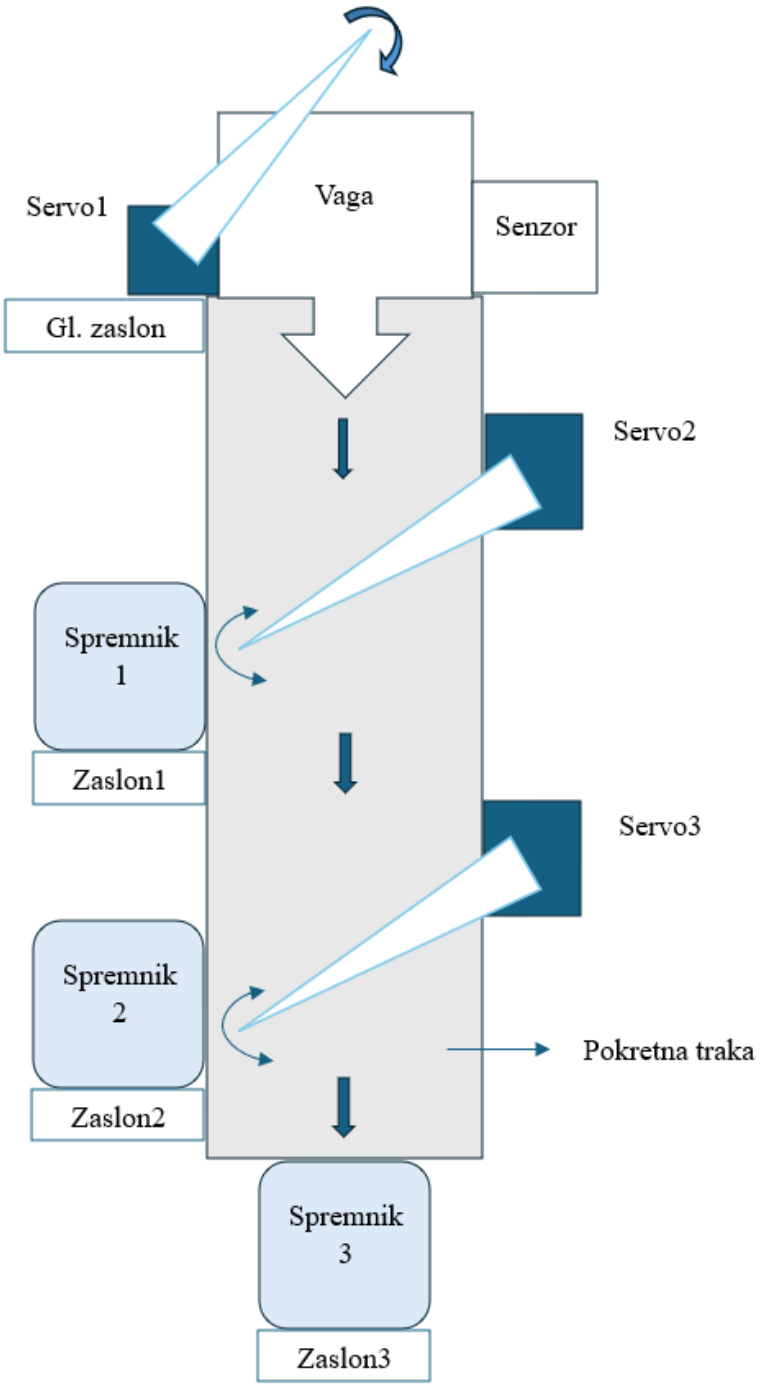


Slika 2.2. *Wheatstoneov most sa otporima naprezanja*[3]



Slika 2.3. *Mjerna ćelija sa označenim otporima naprezanja R1-R4*[4]

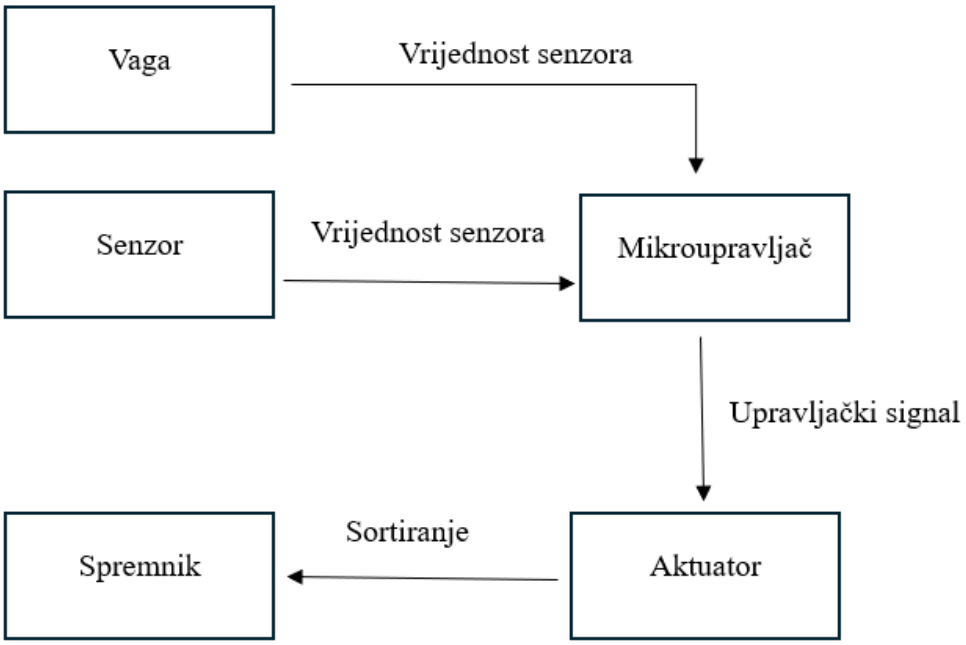
Na slici 2.4. imamo početnu skicu koja prikazuje ugrubo kako bi projekt trebao izgledati. Na slici su prikazani servo motori koji sa svojim pregradama upućuju proizvode prema traci ili u spremnike. Također prikazana je i traka koja transportira proizvode prema spremnicima. Na početku je vaga, na tome mjestu proizvodi stoje i tamo se određuje masa i boja proizvoda. Imamo glavni zaslon koji pokazuje trenutnu masu i boju proizvoda na vagi i zaslone koji prikazuju boju i masu proizvoda u svakom spremniku.



Slika 2.4. Skica makete završnog rada

2.2. Prijedlog sklopovskog rješenja

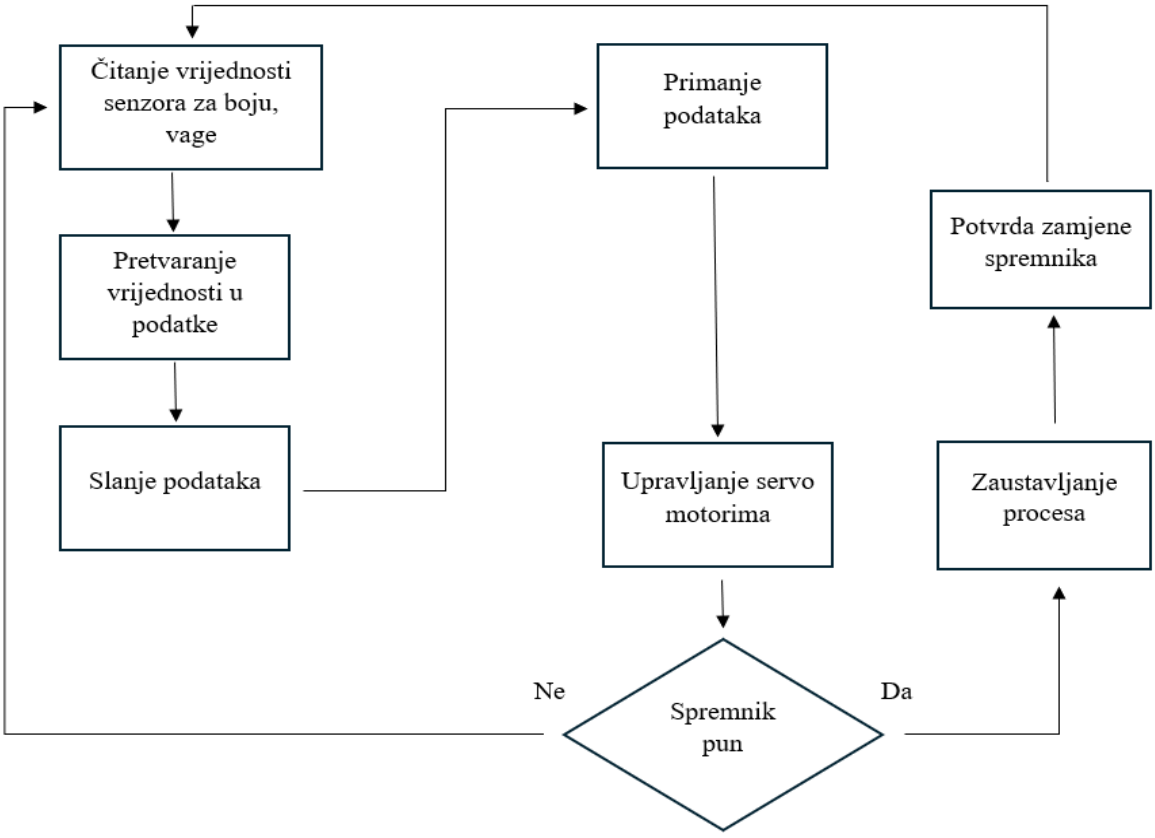
Pokretna traka ima jedan senzor za boju i jednu vagu . Na taj način pomoću senzora za boju i vage detektiramo boju objekta i njegovu masu. Te vrijednosti signala sa senzora i vage se prosljeđuju na mikroupravljač. Mikroupravljač na osnovu tih vrijednosti šalje upravljačke signale i upravlja aktuatorima odnosno servo motorima. Servo motor se rotira i postavlja u definiranu poziciju primljenu s upravljačkog signala mikroupravljača. Objekt pomoću pokretne trake dolazi do zarotiranog servo motora te ga on upućuje u određeni spremnik. Na slici 2.5. imamo blokovsku shemu prijedloga sklopovskog rješenja.



Slika 2.5. *Strukturna shema prijedloga sklopovskog rješenja*

2.3. Prijedlog programskog rješenja

Prijedlog programskog rješenja je realiziran na učinkovit, razumljiv i jednostavan način rada te se izvodi u *Arduino IDE*[5] programu. Počinje se s uključivanjem biblioteka za ekrane, senzor boje, vagu i servo motore. Sve te biblioteke je potrebno definirati kako bi mogli pretvarati signale stanja senzora za boju i vage u razumljive podatke poput boje i vrijednosti mase. Počinjemo s očitavanjem senzora i vage čije signale mikroupravljač prima i pretvara u podatke o boji i masi. Određuje se koja je boja stigla sa senzora za boju te na osnovu toga mikroupravljač šalje upravljačke signale i upravlja određenim servo motorom odnosno postavlja mu kut rotiranja. Za svaki spremnik se definira granica napunjenosti. Dokle god spremnik nije pun proces sortiranja se izvršava. U trenutku kada neki od spremnika dostigne maksimalnu vrijednost napunjenosti zaustavljamo proces. Spremnik je potrebno fizički zamijeniti sa novim te pritiskom na tipku potvrđujemo zamjenu i šaljemo mikroupravljaču signal da je spremnik zamijenjen. Nakon čega se ponovno nastavlja proces sortiranja. Slika 2.6. prikazuje shemu blokovskog prijedloga programskog rješenja.



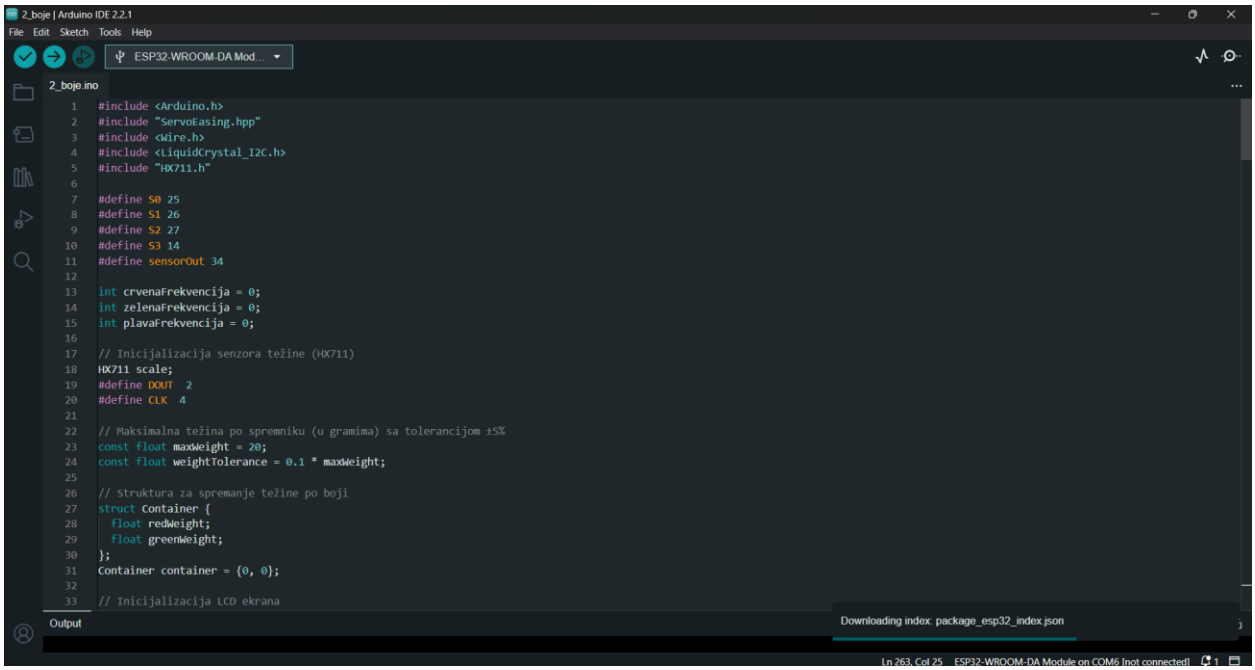
Slika 2.6. *Funkcionalni blok dijagram*

3. REALIZACIJA SUSTAVA ZA SORTIRANJE OBJEKTA

U ovom poglavlju će se govoriti o korištenim programskim alatima i zašto ih koristimo, te će biti objašnjen postupak izrade sklopovskog i programskog rješenja makete.

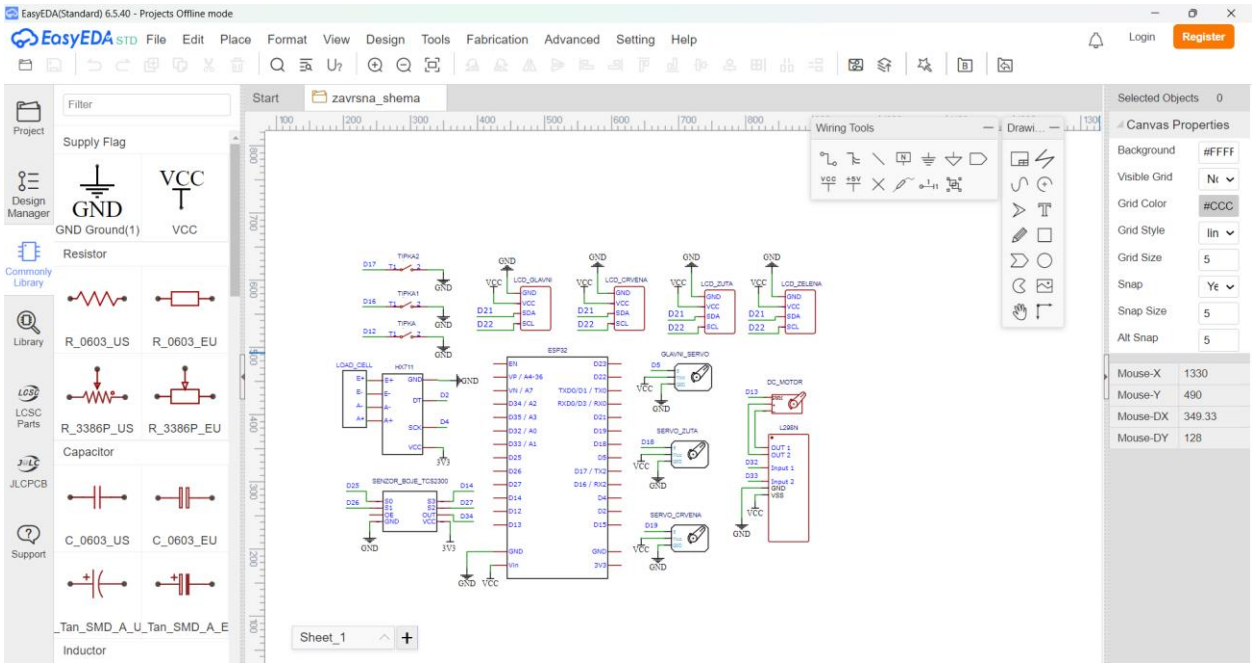
3.1. Korišteni komponente, alati i programska okruženje

Za pisanje koda se koristio ArduinoIDE programski alat. Omogućuje programiranje mikroupravljača i svih ostalih korištenih komponenti. Njegova rasprostranjenost i učinkovitost se pokazala kao najbolja opcija za izvršavanje programskog dijela rada. Programski kod se nalazi u prilogu.[7]



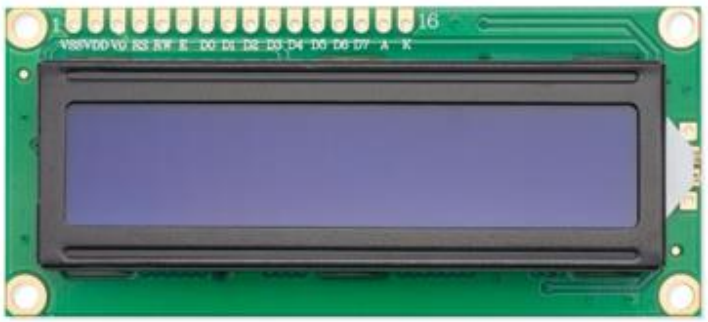
Slika 3.1. *prikaz Arduino IDE programa*

Pri kreiranju sklopovskog rješenja koristio se *EasyEDA*[6] program. *EasyEDA* je besplatan i prilagođen je za studente, inženjere. Koristili smo ga kako bi prikazali električnu shemu spajanja komponenti makete.



***Slika 3.2**. prikaz EasyEDA programa*

Za prikaz informacija u obliku teksta koristimo *LCD 16x2*[8] ekrane. Prikazuje 16 znakova u 2 reda. Ekran je popularan radi svoje rasprostranjenosti i jednostavnosti programiranja, spajanja.

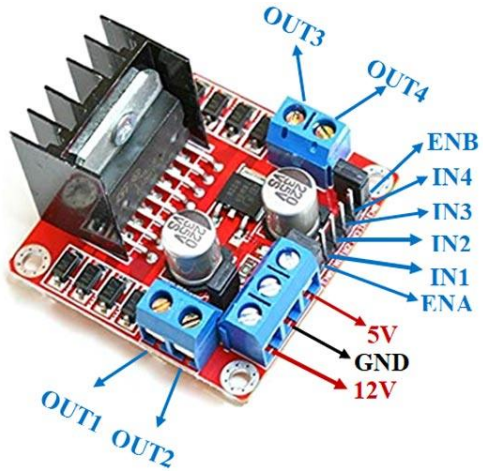


***slika 3.3.** LCD 16x2 ekran*[8]

MG996R [9] je servomotor koji je iznimno popularan zbog svojih performansi, stabilnosti i snage. Stekao je svoju popularnost zbog svoje raznovrsne primjene. U ovome radu koristimo tri takva servomotora jedan za guranje predmeta s vage na pokretnu traku, a druga dva za upućivanje predmeta s pokretne trake u odgovarajući spremnik.

Slika 3.4. *MG996R servo*[9]

Koristimo *L298N drive[10]* koji nam služi za prijenos napajanja na motor te samu kontrolu rada motora odnosno vrtnju pokretne trake. Motor možemo paliti i gasiti preko *L298N drivea* davajući odgovarajući digitalni signal 0 ili 1.



Slika 3.5. *L298N drive*

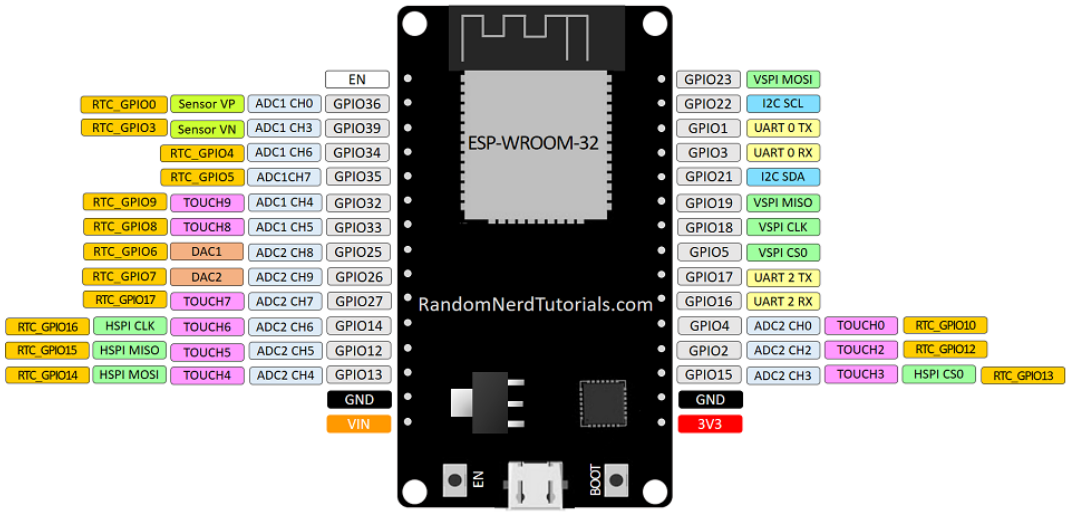
Za detekciju boje koristimo *TCS3200* senzor boje[11]. On pretvara svjetlost u frekvencijske signale. Sadrži foto diode koje su osjetljive na različite boje točnije na crvenu, zelenu i plavu. Sastoji se i od filtera koji propuštaju samo jednu od tih boja i blokiraju ostale boje. Kada nam se aktivira određeni filtar, foto diode detektiraju intenzitet svjetlosti te boje. Nakon toga senzor mjeri intenzitet te boje i pretvara tu vrijednost u digitalni signal. Na temelju tih vrijednosti moguće je odrediti boju predmeta ispred senzora.

Slika 3.6. *senzor boje TCS3200*

Kako bi saznali iznos mase svakog objekta na ulazu makete koristimo se s *Load-cellom*(mjerna ćelija*)*[2]. On nam pretvara vrijednost sile obično težine ili pritiska u električni signal. Taj električni signal je proporcionalan djelovanju sile i šalje se mikroupravljaču.

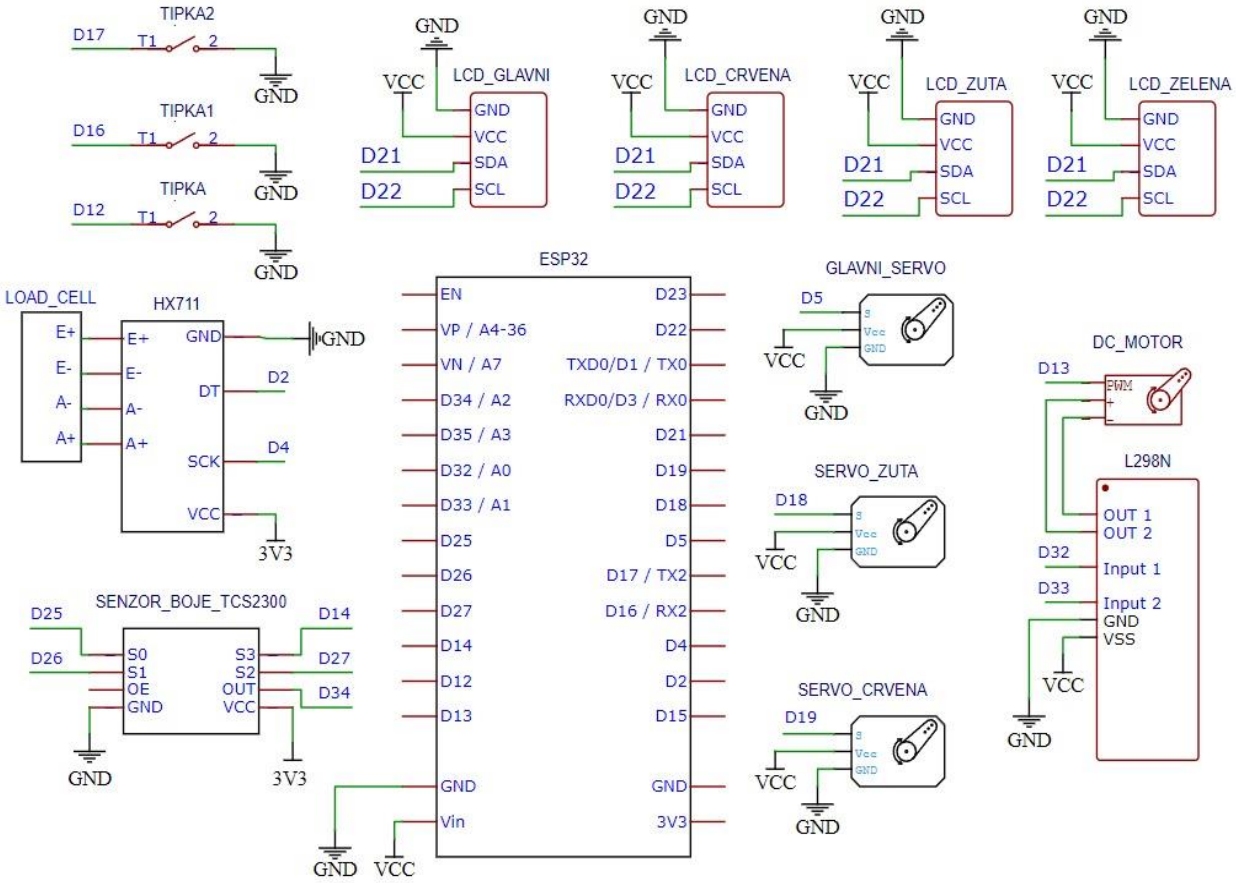
Slika 3.7. *load cell(mjerna ćelija)*

ESP32-WROOM[12] je mikroupravljački modul baziran na ESP32 čipu. Ima nisku potrošnju energije i brojne digitalne i analogne ulaze/izlaze koji omogućuju povezivanje senzora, aktuatora i drugih komponenti. Za maketu se koristi radi svoje fleksibilnosti, lakog programiranja i mogućnosti obavljanje više zadataka odjednom. Sve to ga čini odličnim izborom za ovaj rad.



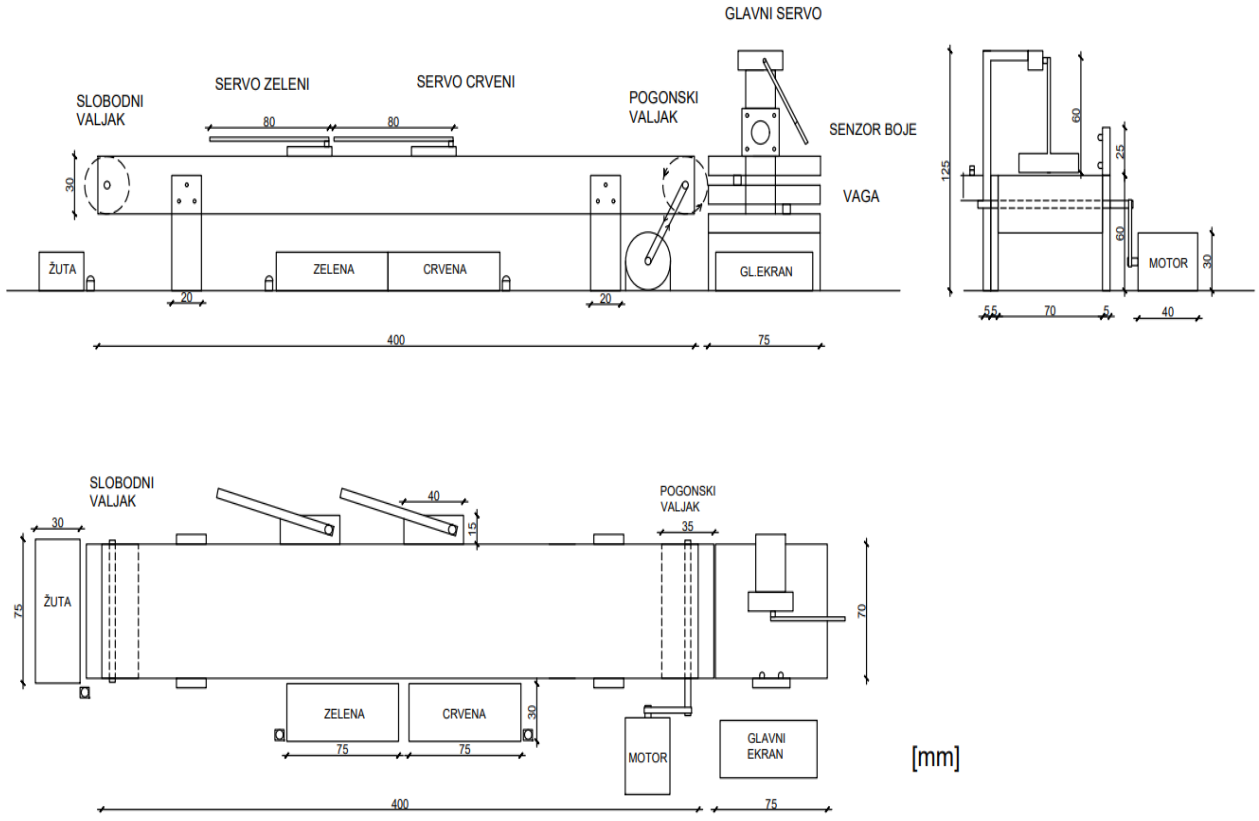
Slika 3.8. *ESP32-WROOM s popisom pinova*

3.2. Realizacija konstrukcijskog i sklopovskog rješenja

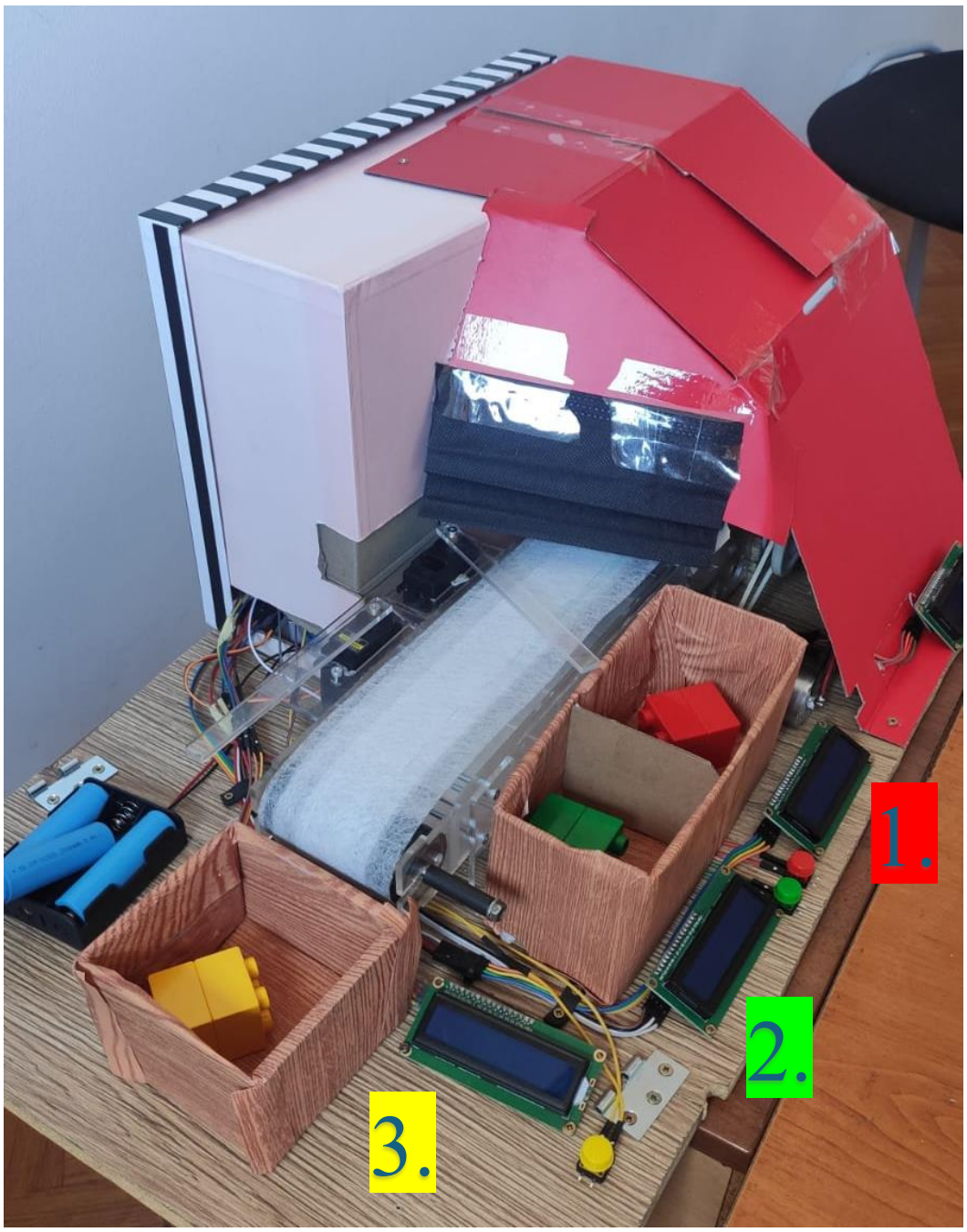


Slika 3.9. *električna shema*

Na električnoj shemi (slika 3.9) prikazano je spajanje svih komponenti potrebni za realizaciju sustava koji mjeri masu i detektira boju. Kontrolu rada motora pokretne trake reguliramo pomoću L298N drivea koji nam služi za njegovo pokretanje i zaustavljanje. Za napajanje LCD ekrana, servo motora i L298N drivea smo koristili vanjski izvor od 5V, a za load cell i senzor boje 3.3V s mikroupravljača. Sve komponente nam imaju zajedničko uzemljenje. Uz pinove za napajanje i uzemljenje imamo upravljačke pinove na komponentama koje spajamo na ESP32 prema električnoj shemi (3.3.). Tipkala žute,crvene i zelene boje koja služe za resetiranje vrijednosti spremnika na LCD ekrana spojena su na pinove D17, D16 i D12 . Pin SDA s LCD ekrana spajamo na D21 dok pin SCL na D22. Pin za upravljački signal servo motora spajamo na D5 pin za glavni servo, D18 za žuti servo i D19 za crveni servo. Motor za pokretanje trake spajamo na D13 pin te na OUT1 i OUT2 s L298N drivea. L298N drive spajamo na D32 za Input1 te D33 za Input2.



Slika 3.10. *2d nacrt*



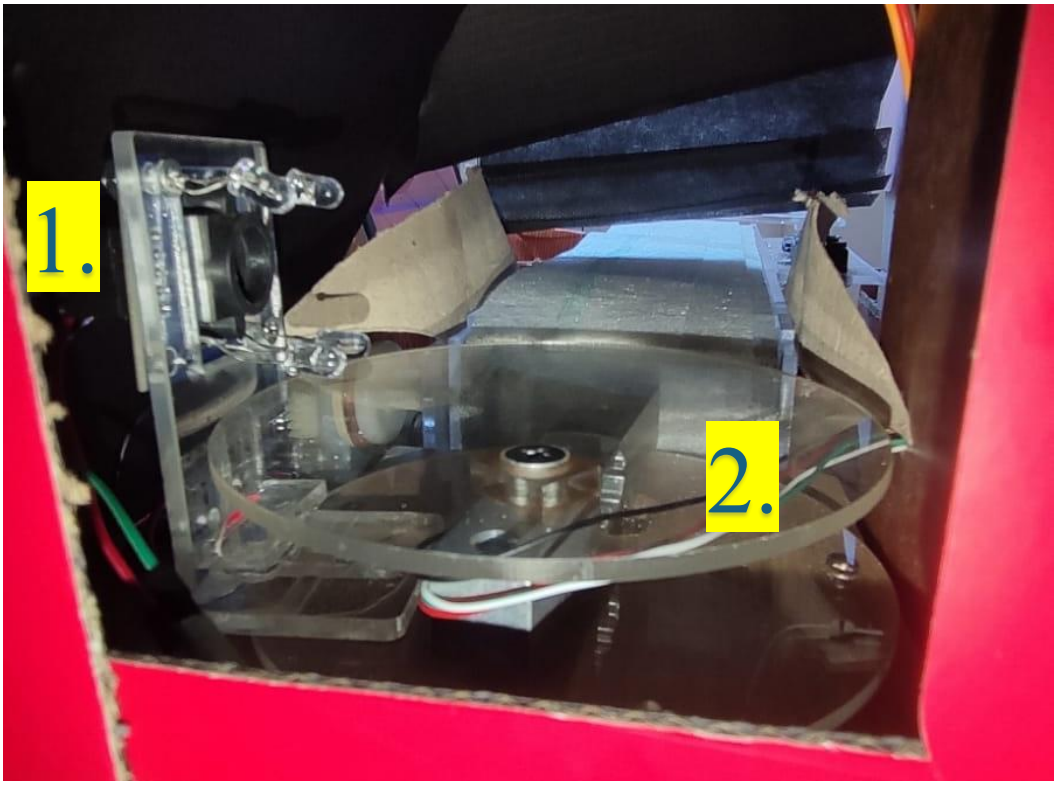
Slika 3.11. *završni izgled makete*

1- Spremnik, ekran i tipka za crvenu boju
2- Spremnik, ekran i tipka za zelenu boju
3- Spremnik, ekran i tipka za žutu boju

Slika 3.12. *prikaz s gornje strane*

Slika 3.13. *prikaz s prednje strane*

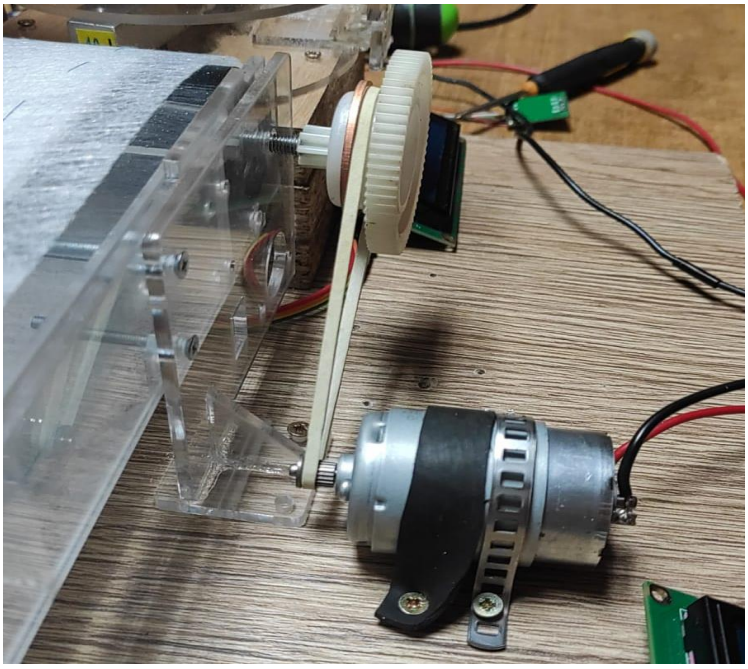
1- Glavni ekran



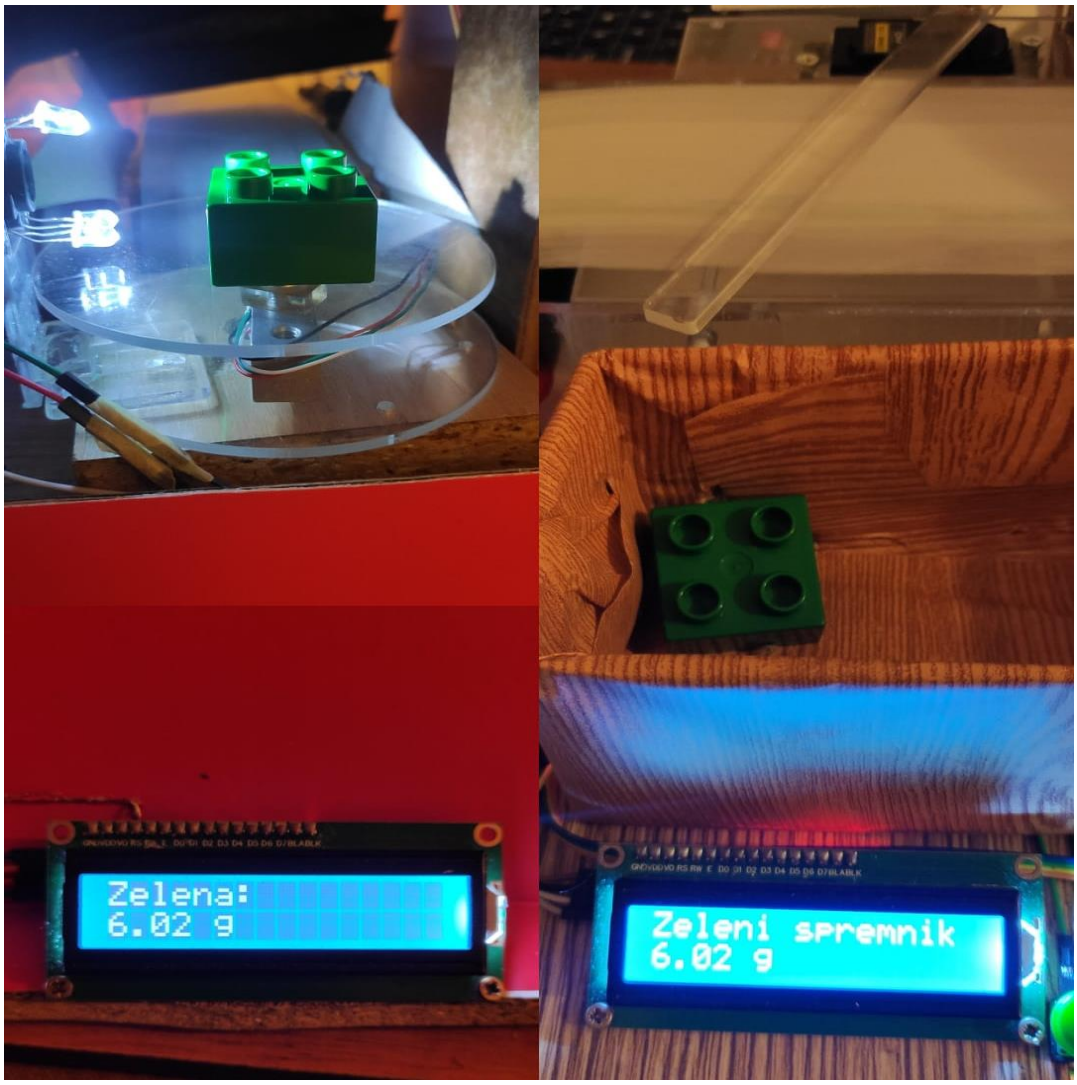
Slika 3.14. *otvor za postavljanje objekta*

1- Senzor boje

2- Vaga

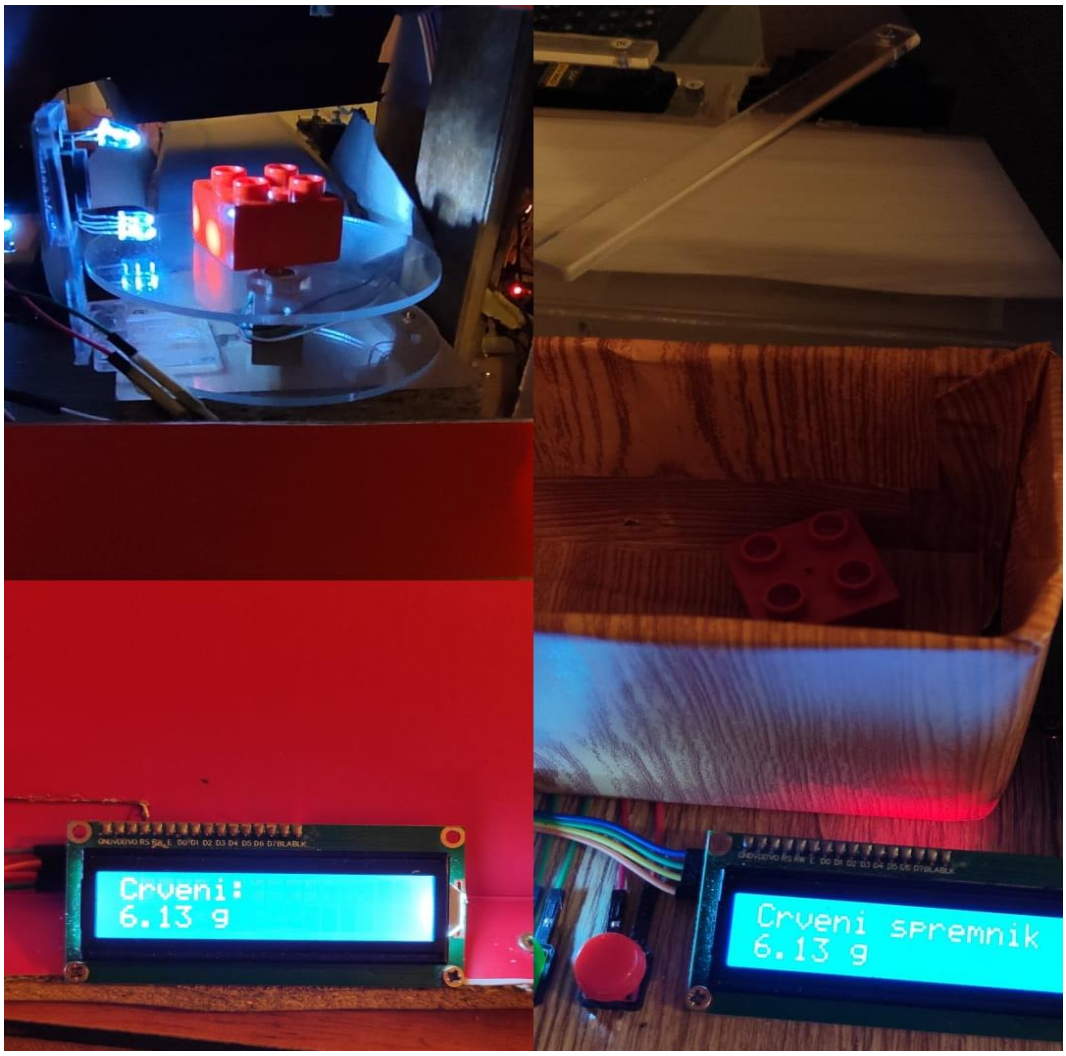


Slika 3.15. *motor povezan na pogonski valjak*



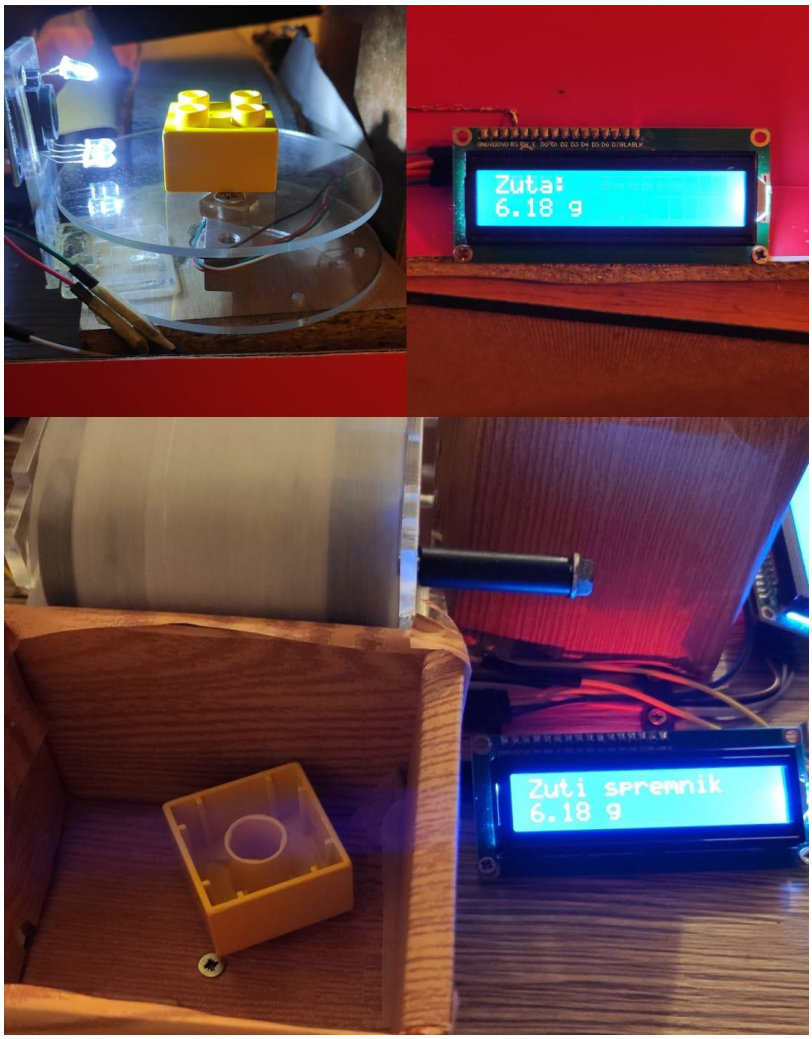
Slika 3.16. *prikaz sortiranja zelenog objekta*

Slika 3.16. prikazuje cjelokupan proces sortiranja zelenog objekta. Postavljamo ga na vagu gdje mu se određuje masa i detektira boja. Na glavnom ekranu dobijemo ispis o njegovoj boji i masi u ovom slučaju „Zelena: 6.02g“ te nam se aktivira servo za zelenu boju i postavlja pregradu koja upućuje objekt u spremnik za zelenu boju. Nakon toga dobivamo ispis na ekranu za spremnik zelene boje s iznosom trenutne mase u spremniku.



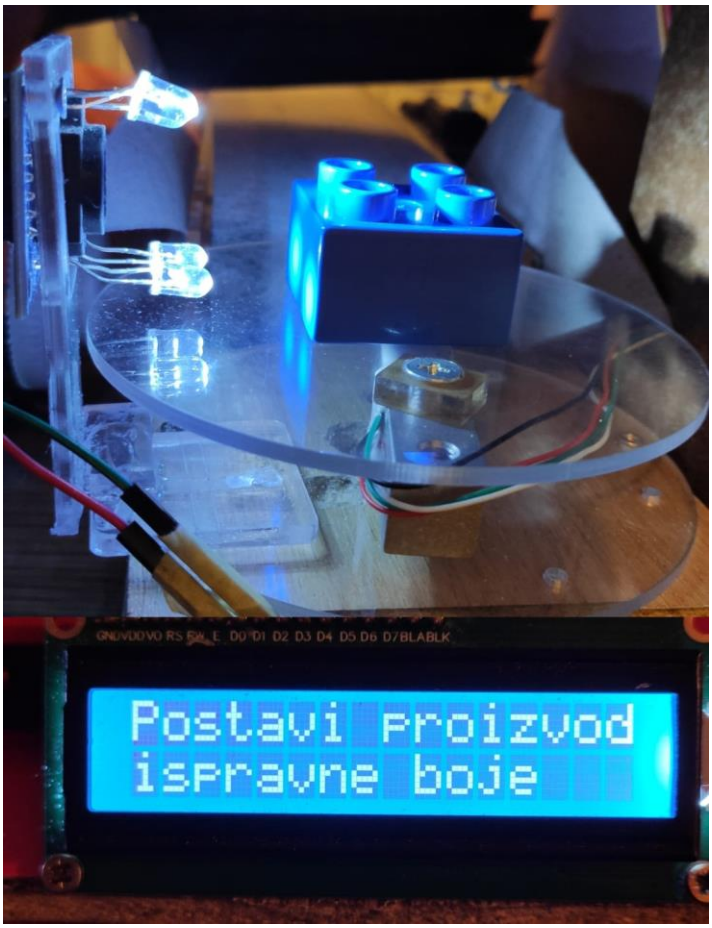
Slika 3.17. *prikaz sortiranja crvenog objekta*

Slika 3.17. prikazuje cjelokupan proces sortiranja crvenog objekta. Postavljamo ga na vagu gdje mu se određuje masa i detektira boja. Na glavnom ekranu dobijemo ispis o njegovoj boji i masi u ovom slučaju „Crveni: 6.13g“ te nam se aktivira servo za crvenu boju i postavlja pregradu koja upućuje objekt u spremnik za crvenu boju. Nakon toga dobivamo ispis na ekranu za spremnik crvene boje s iznosom trenutne mase u spremniku.



Slika 3.18. *prikaz sortiranja žutog objekta*

Slika 3.18. prikazuje cjelokupan proces sortiranja žutog objekta. Postavljamo ga na vagu gdje mu se određuje masa i detektira boja. Na glavnom ekranu dobijemo ispis o njegovoj boji i masi u ovom slučaju „Žuta: 6.18g“ te nam se servo motori za crvenu i zelenu boju postavljaju u početnu poziciju kako bi propustili objekt slobodno preko cijele pokretne trake gdje se na kraju nalazi spremnik. Nakon toga dobivamo ispis na ekranu za spremnik žute boje s iznosom trenutne mase u spremniku.



Slika 3.19. *prikaz detekcije krive boje*

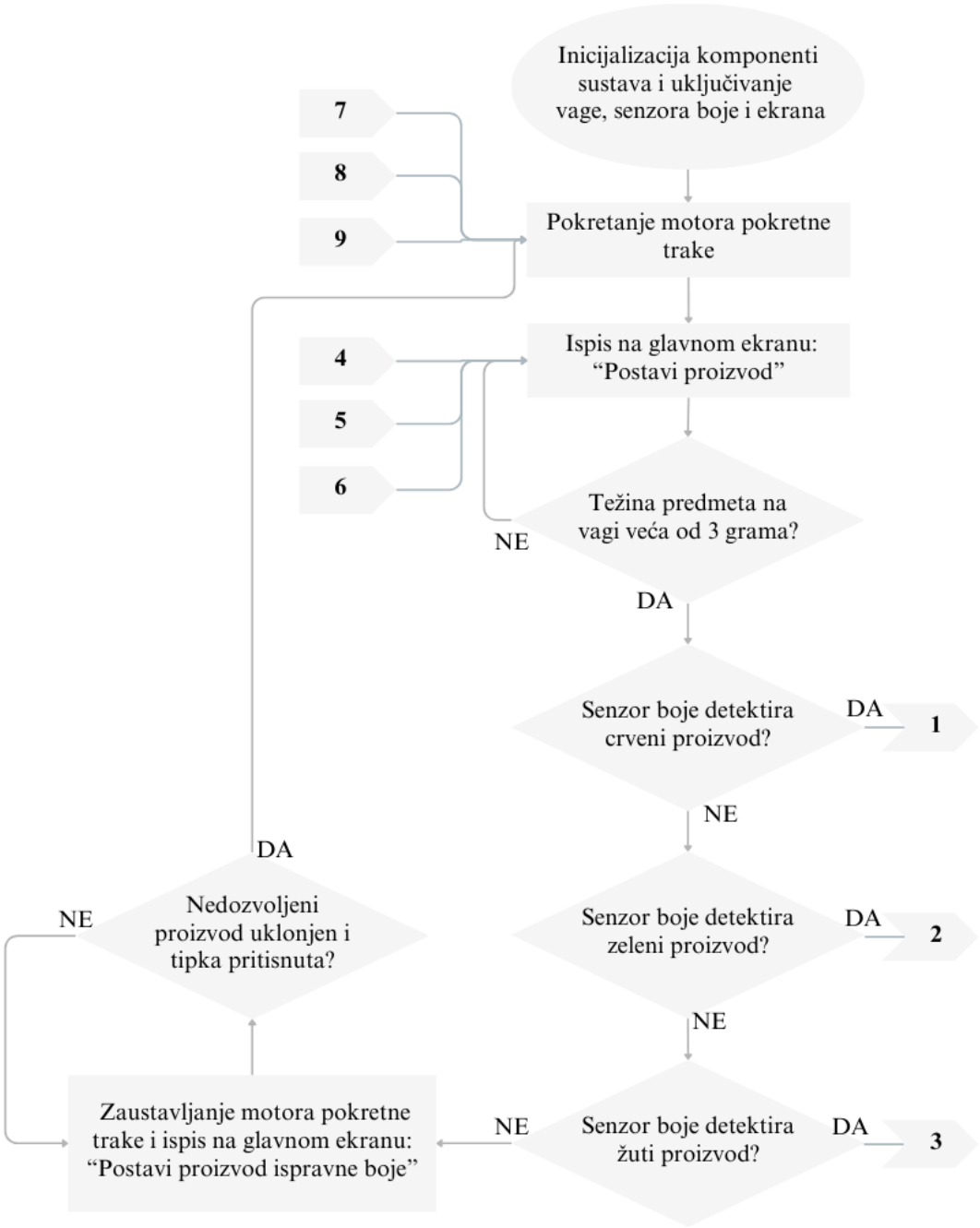
Ako postavimo objekt neke boje koja nam ne spada pod crvenu, žutu ili zelenu u ovom slučaju nam je to plava. Proces sortiranja će se zaustaviti te ćemo na glavnom ekranu dobiti ispisanu prikazanu poruku „Postavi proizvod ispravne boje“.

Slika 3.20. *prikaz ekrana pri punim spremnicima*

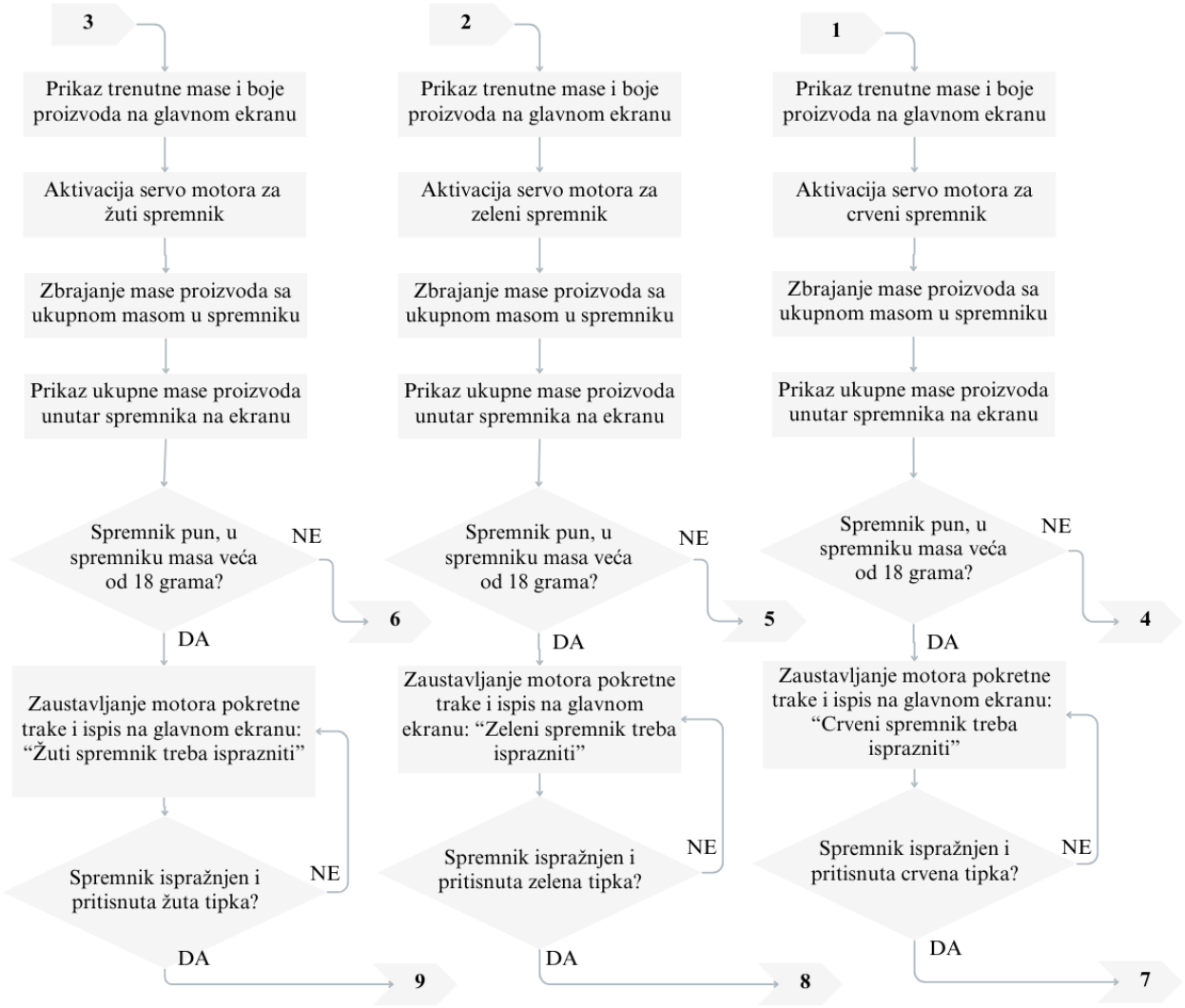
Kada nam se spremnik u nekoj boji napuni, primjer crveni na ekranu od tog spremnika dobijemo poruku „Crveni pun“ dok na glavnom ekranu dobijemo poruku „Crveni spremnik treba isprazniti“. Iste poruke dobijemo za žutu i zelenu boju što možemo vidjeti iz priložene slike 3.20..

3.3. Realizacija programskog rješenja

Dijagram toka gotovog projekta je malo drugačiji u odnosu na pretpostavljenu, teorijsku obradu programskog rješenja. Glavni princip rada sustava za sortiranje je očitavanje boje, ovisno o očitanoj boji u sustavu se servo motori postavljaju na različite pozicije i time guraju proizvode u određene spremnike. Sustav također radi i na principu mjerenja mase kako bi se očitala popunjenost spremnika kao što je prikazano na toku rada na slikama dolje (slike 3.21. i 3.22.).



Slika 3.21. *dijagram toka 1*



Slika 3.22. *dijagram toka 2*

Nakon što se program pokrenuo, motor kreće rotirati pokretnu traku, na glavnom LCD ekranu dobivamo poruku kako možemo krenuti sa postavljanjem predmeta na vagu. Ukoliko je predmet teži od 3 grama, program očitava boju predmeta, a na glavnom ekranu se prikazuje boja postavljenog predmeta zajedno sa izmjerenom masom. Nakon detekcije boje i mase predmeta servo motor iznad vage pomiče predmet prema pokretnoj traci, iznad koje postoje drugi servo motori koji svojim pozicioniranjem usmjeravaju predmete u predviđene spremnike. Prvi servo motor preusmjerava prema crvenom spremniku, drugi prema zelenom spremniku, ukoliko je predmet žute boje – servo motori izmiču svoje pregrade kako bi proizvod mogao bez ometanja, ravno, nošen trakom, doći do svog spremnika koji se nalazi na dnu pokretne trake. Program broji mase svakog sortiranog predmeta unutar spremnika, te izbacuje poruku kada je spremnik pun, tj.

kada se unutar spremnika nalazi 18 grama predmeta. Poruka o napunjenosti spremnika dolazi na LCD ekran punog spremnika i na glavni LCD ekran se ispisuje da ga treba isprazniti. Kako bi program detektirao pražnjenje spremnika u sustavu smo dodali tipke. Postoje žuta, zelena i crvena tipka koja svaka resetira stanje u svojem spremniku. Ukoliko se u sustavu očita predmet koji je neke četvrte boje koju program ne prepoznaje, motor pokretne trake se zaustavlja i tek nakon što se makne predmet sa vage i pritisne bilo koja tipka program nastavlja sa daljnjim radom.

4. TESTIRANJE I REZULTATI

4.1. Metodologija testiranja

Kako bi se moglo pouzdati u izgrađeni sustav, kako bi osigurali pouzdan rad određivanja boje, mase i pravilno sortiranje u predviđen spremnik, treba testirati proces. Obavili smo više testiranja programa : test i podešavanje rada senzora boje, testiranje pravilne detekcije mase proizvoda i testiranje sortiranja predmeta u predviđeni spremnik.

4.2. Rezultati testiranja

4.2.1. Test i podešavanje rada senzora boje

Za testiranje i pravilno podešavanje senzora boje koristimo *serial monitor* (serijsku komunikaciju) unutar računala na kojeg je spojen mikroupravljač u kojem se ispisuje određena boja sa senzora. Potrebno je odrediti udaljenost od senzora boje koja je dovoljno blizu da se pravilno odredi boja proizvoda, ali ne preblizu kako nebi proizvod zapeo za senzor ili promašio pokretnu traku. Potrebno je namještati proizvod na vagi da ima dovoljnu udaljenost od senzora kako bi se očitala točna boja. Za testiranje koristimo plastičnu kockicu koja predstavlja predmet crvene boje.

Potrebno je testirati koja je minimalna udaljenost pri kojoj se detektira boja predmeta, tj. pri kojoj se udaljenosti unutar *serial monitora* ispiše da je proizvod crvene boje. Srednjom vrijednošću više mjerenja minimalne udaljenosti prepoznavanja boje ćemo postaviti granice koje govore kako postaviti predmet naspram senzora. Kako se na senzoru boje nalaze LED diode, predmet možemo postaviti točno ispred njih što je minimalna udaljenost od 10mm, pa od te vrijednosti krećemo dok točno ne očitamo boju.

Mjerenje	Minimalna udaljenost [mm]	Maksimalna udaljenost [mm]
1	14	25
2	14	24
3	15	25
4	13	26
5	14	24
6	15	26
7	13	25
8	16	24
9	13	26
10	15	25
11	13	25
12	14	24
13	13	26
14	14	25
15	18	25
16	15	26
17	14	24
18	14	24
19	14	24
20	15	25

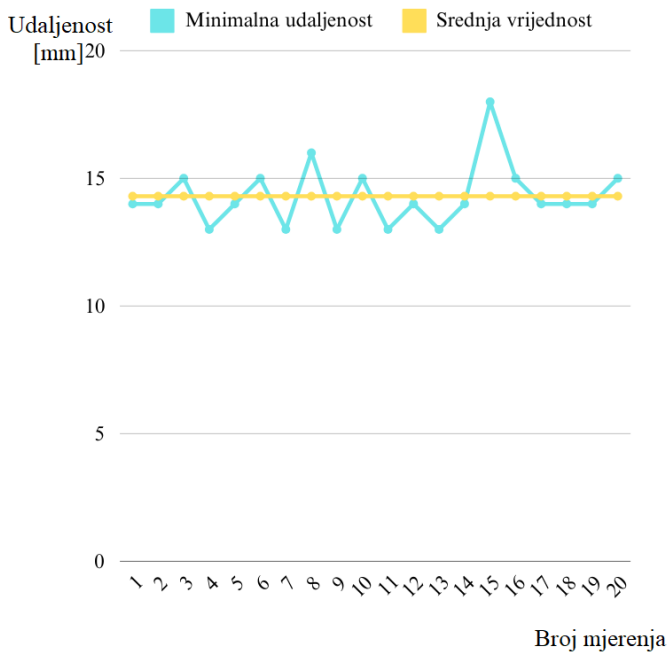
Tablica 4.1. *Testiranje minimalne udaljenosti detekcije boje*

Srednja vrijednost je dobivena iz formule:

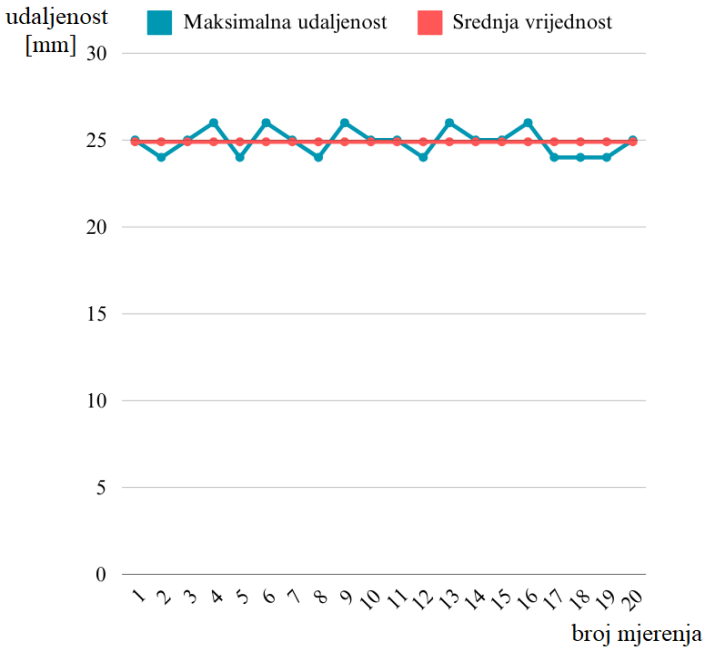
$$\text{sr.vr.} = \frac{\text{zbroj izmjerenih veličina}}{\text{broj mjerenja}}$$

Srednja minimalna vrijednost :	14.3 mm
Srednja maksimalna vrijednost :	24.9 mm

Tablica 4.2. *Izračun srednjih vrijednosti minimalne i maksimalne udaljenosti*



Slika 4.3. *Graf testiranja minimalne udaljenosti*



Slika 4.4. *Graf testiranja maksimalne udaljenosti*

Iz tablice, grafova i izračuna srednje vrijednosti možemo uočiti da je minimalna granica očitanja boje dosegla 13 milimetara, te da je srednja vrijednost minimalne udaljenosti potrebna za detekciju proizvoda 14.3 milimetra. Maksimalna granica očitanja je dosegla 26 milimetara, a srednja vrijednost maksimalne udaljenosti potrebna za detekciju je 24.9 milimetara. Kako bi

detekciju obavili što pravilnije, predmet ipak treba postavljati što bliže sredini vage, malo dalje od minimalne i maksimalne udaljenosti.

4.2.2. Testiranje pravilne detekcije mase proizvoda

Kako se program bazira na mjerenju mase proizvoda kako bi se očitala napunjenost spremnika, bitno je da ovaj dio sustava što bolje funkcionira. Za testiranje proizvoda koristimo kockice od plastike težine 6 grama koje postavljamo na isti način na vagu. Mjerimo 20 puta, a glavni LCD ekran nam ispisuje vrijednosti izmjerene sa vage nakon što se prepozna predmet teži od 3 grama na vagi.

Mjerenje	Masa	Mjerenje	Masa
1	6,03 g	11	5,82 g
2	5,92 g	12	6,03 g
3	6,11 g	13	6,21 g
4	5,86 g	14	6,08 g
5	6,07 g	15	5,87 g
6	5,89 g	16	5,91 g
7	6,01 g	17	6,17 g
8	5,97 g	18	6,1 g
9	6,04 g	19	6,15 g
10	5,98 g	20	5,92 g

Tablica 4.5. *Testiranje detekcije mase proizvoda*

Srednja vrijednost :	6.007 g
Najveće odstupanje :	+0.21 g

Tablica 4.6. *Izračun srednje vrijednosti i odstupanja mase proizvoda*

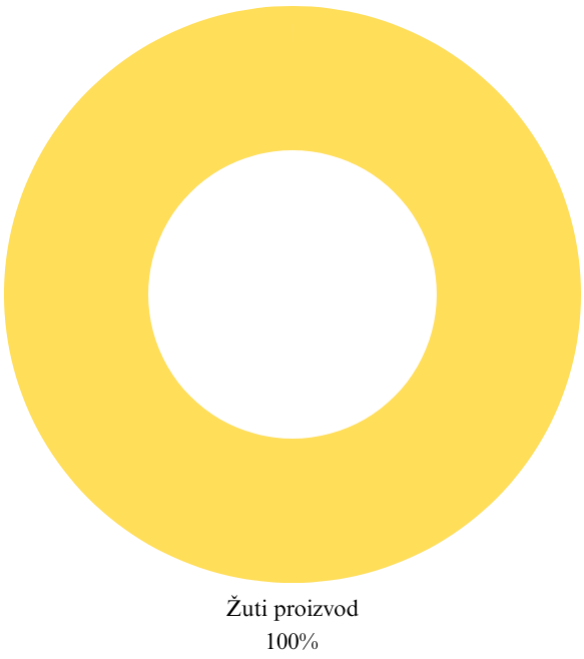
Srednja vrijednost je dobivena iz formule: $sr.vr. = \frac{\text{Zbroj izmjerenih veličina}}{\text{Broj testiranja}}$

Mjerno odstupanje je Δm = rezultat mjerenja – prava vrijednost.

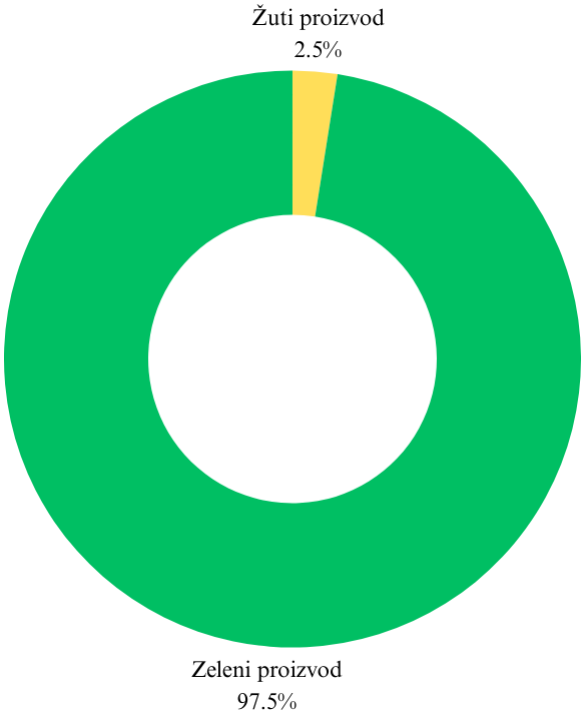
Najveće mjerno odstupanje je od 0.21 gram, a najmanje 0.01 gram. Na to su odgovorni vanjski utjecaji i mjerna osjetljivost vage. Uočavamo da mjerenjem više puta i izračunom srednje vrijednosti dolazimo bliže stvarnoj vrijednosti.

4.2.3. Testiranje sortiranja predmeta u predviđeni spremnik

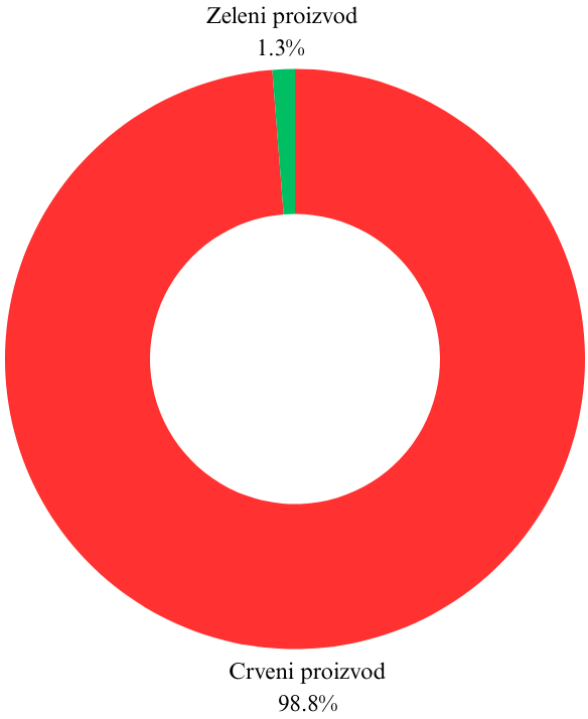
Ovaj test služi za testiranje cijelog sustava, servo motora i timera koji upravljaju motorima. Koristimo plastične kockice crvene, žute i zelene boje te ih postavljamo u sustav kako bi se sortirale u svoje predviđene spremnike. Na slijedećim grafovima vidimo postotak uspješnosti sortiranja svakog spremnika u 80 mjerenja, postavljamo žuti proizvod, pa zeleni i onda crveni – i tako 80 puta. U žuti spremnik kocke idu ravno po pokretnoj traci, servo motori se izmiču kako nebi smetali predmetu na traci. U zeleni spremnik predmet dolazi kada se uključi drugi servo motor koji usmjerava zelene predmete prema spremniku. Zbog prebrzog postavljanja novog predmeta dogodilo se da je i žuti predmet poguran u zeleni spremnik jer se servomotor za novi proizvod aktivirao, a stari proizvod nije stigao do svog spremnika. Isto vrijedi i za crveni spremnik. Zaključujemo da je potrebno sporije postavljati proizvode na vagu, ili usporiti program kako bi omogućio siguran dolazak predmeta u određene spremnike.



Slika 4.7. *graf predmeta u žutom spremniku*



Slika 4.8. *graf predmeta u zelenom spremniku*



Slika 4.9. *graf predmeta u crvenom spremniku*

5. ZAKLJUČAK

Ovim završnim radom smo teorijski zamislili, fizički napravili i u kodu programirali pokretnu traku sa sustavom za sortiranje predmeta po boji i masi. Cilj je bio da se korištenjem senzora boje očita boja i da se kretajući trakom gura do svoga određenog spremnika. Korištenjem vage na ulazu u sustav smo omogućili praćenje mase svakog predmeta, time smo omogućili kontrolu napunjenosti spremnika. Pokretna traka je napravljena od plastičnih materijala, gume, tekstila i vijaka, uz pomoć kartona smo omogućili zamračivanje sustava kako bi se što bolje očitavala boja. Sustav je realiziran uz povezivanje mikroupravljača sa servomotorima, senzorom boje, vage i tipkala. Kako bi izradili projekt pomoglo nam je fakultetsko obrazovanje kao što je programiranje, fizika, elektronika, elektrotehnika.

Projekt je izrađen u sklopu projektnog zadatka za predmet mikroračunala u automatizaciji. Naspram početne verzije projekta, završna verzija je dosta kompliciranija, dodali smo dodatnu boju unutar sustava koja se sortira. Tijekom izrade smo mijenjali fizičke karakteristike, pokrivali smo i pomicali senzor boje unutar sklopa kako bi se zaštitio od svjetlosti, mijenjali smo servo motore kako bi pouzdanije radili i mijenjali dijelove koji su bili izrađeni od kartona sa kvalitetnijim materijalom kao što je plastika.

Izrađeni rad funkcionira kako je zamišljeno, ali ima svojih nedostataka. Najveći nedostatak je brzina pri kojoj se sortiraju predmeti. Trenutno se postavlja jedan po jedan predmet na traku, kada bi se uspostavio sustav sa još jednom trakom koja dovodi proizvode na vagu kako bi se sortirali, ne bi bilo potrebna ljudska ruka za postavljanje proizvoda – samo bi praznili spremnike kada bi to bilo potrebno. Proces bi mogao biti više precizan korištenjem preciznijih senzora boje, ili kamera, jer bi se točnije mogle očitavati boje proizvoda. Također korištenjem preciznije, skuplje vage bi došlo do boljih rezultata, ne bi bilo nepravilnosti pri sortiranju. Kada bi se cijeli proces povećao u svim dimenzijama naš projekt bi mogao biti korišten za druge različite upotrebe unutar poljoprivrednih plodova, a i šire.

LITERATURA

[1] Sensorbasic color info, https://www.keyence.com/Images/sensorbasics_color_info_img_01_1838367.png , 26.06.2024

[2] How Does a Strain Gauge Work?, https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html , 26.06.2024

[3] How Does a Strain Gauge Work?, https://www.800loadcel.com/assets/components/phpthumbof/cache/wheatstone%20bridge%202.d20c140a08358af779d38cf1285ff6c4.webp , 26.06.2024

[4] How Does a Strain Gauge Work?, https://www.800loadcel.com/assets/components/phpthumbof/cache/strain%20gauge%202b.d90f5871adda955748438f17b28d07c6.webp , 26.06.2024

[5] Arduino IDE info,

https://docs.arduino.cc/software/ide/#ide-v2 , 26.06.2024

[6] EasyEDA info,

https://easyeda.com/ , 28,06,2024

[8] LCD 16x2 ekran

https://www.winstar.com.tw/products/character-lcd-display-module/16x2-lcd.html , 28.06.2024

[9] MG996R servo,

https://www.towerpro.com.tw/product/mg996R/ , 28.06.2024

[10] L298N drive,

https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/ , 29.06.2024

[11]Color sensor,

https://www.waveshare.com/wiki/Color_Sensor , 27.08.2024

[12] ESP32-WROOM mikroupravljač,

https://www.nabto.com/guide-to-iot-esp-32/ , 27.08.2024

SAŽETAK

Naslov: Sustav za sortiranje objekata s primjenom u poljoprivrednoj proizvodnji

Cilj ovog završnog rada bio je osmisliti, dizajnirati, izraditi i testirati pokretnu traku sa sustavom za sortiranje objekata prema boji i zbrajanje njihove mase, te njihovo raspoređivanje u odgovarajuće spremnike. Napravljen je osvrt na povijest pokretnih traka te analizirane prednosti i nedostatci njihove automatizacije. Objašnjen je teorijski koncept rada trake, kao i hardversko te softversko rješenje. Projekt se temelji na ESP32 mikrokontroleru koji je programiran uz pomoć Arduino razvojnog okruženja. Cijeli sustav je podijeljen u tri dijela: pokretnu traku, mjerne stanice i stanice za sortiranje predmeta.

Ključne riječi **:** ESP32, Arduino, pokretna traka, sustav za mjerenje mase i detekciju boje, sortiranje, mikrokontroler.

ABSTRACT

Title: System for object sorting in agriculture.

The goal of this thesis was to design, develop, and test a conveyor belt system capable of sorting objects by color and calculating their mass, then distributing them into appropriate containers. It includes a review of the history of conveyor belts and an analysis of the advantages and disadvantages of their automation. The theoretical concept of the conveyor's operation, as well as the hardware and software solutions, are explained. This thesis is based on the ESP32 microcontroller, which is programmed using the Arduino development environment. The entire system is divided into three parts: the conveyor belt, measuring stations, and sorting stations.

Keywords: ESP32, Arduino, conveyor belt, mass measurement and color detection system, sorting, microcontroller

ŽIVOTOPIS

Denis Crnoja rođen 07. veljače 2000. godine u Sisku. Pohađa Osnovnu školu Zvonimira Franka u Kutini od 2006. godine. Nastavlja se obrazovati u Tehničkoj školi Kutina 2014. godine. Nakon završetka upisuje stručni studij elektrotehnike smjer Automatika na Fakultetu elektrotehnike i računarstva i informacijskih tehnologija u Osijeku 2018. godine.

PRILOZI

[7] Programski kod

```
#include <Arduino.h>
#include "ServoEasing.hpp"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "HX711.h"

#define S0 25
#define S1 26
#define S2 27
#define S3 14
#define sensorOut 34

int crvenaFrekvencija = 0;
int zelenaFrekvencija = 0;
int plavaFrekvencija = 0;


// Inicijalizacija senzora težine (HX711)
HX711 scale;
#define DOUT  2
#define CLK  4


// Maksimalna težina po spremniku (u gramima) sa tolerancijom ±5%
```

```
const float maxWeight = 20;
const float weightTolerance = 0.1 * maxWeight;

// Struktura za spremanje težine po boji
struct Container {
  float redWeight;
  float greenWeight;
  float yellowWeight;
};
Container container = {0, 0};

// Inicijalizacija LCD ekrana
LiquidCrystal_I2C lcdCurrent(0x27, 16, 2);  // Za trenutnu boju i masu
LiquidCrystal_I2C lcdRed(0x25, 16, 2);      // Za crveni spremnik
LiquidCrystal_I2C lcdGreen(0x26, 16, 2);    // Za zeleni spremnik
LiquidCrystal_I2C lcdYellow(0x22, 16, 2);    // Za zuti spremnik

// Funkcije za prikaz težine
void checkWeightRed(float weight) {
  lcdRed.clear();
  lcdRed.setCursor(0, 0);
  if (weight >= maxWeight - weightTolerance) {
    lcdRed.print("Crveni pun");  // Display "Spremnik pun" za crvenu
  } else {
    lcdRed.print("Crveni spremnik");
    lcdRed.setCursor(0, 1);
    lcdRed.print(weight);
    lcdRed.print(" g"); } }

void checkWeightGreen(float weight) {
```

```
  lcdGreen.clear();
  lcdGreen.setCursor(0, 0);
  if (weight >= maxWeight - weightTolerance) {
    lcdGreen.print("Zeleni pun");  // Display "Spremnik pun" za zelenu
  } else {
    lcdGreen.print("Zeleni spremnik");
    lcdGreen.setCursor(0, 1);
    lcdGreen.print(weight);
    lcdGreen.print(" g"); } }

void checkWeightYellow(float weight) {
  lcdYellow.clear();
  lcdYellow.setCursor(0, 0);
  if (weight >= maxWeight - weightTolerance) {
    lcdYellow.print("Zuti pun");  // Display "Spremnik pun" za zelenu
  } else {
    lcdYellow.print("Zuti spremnik");
    lcdYellow.setCursor(0, 1);
    lcdYellow.print(weight);
    lcdYellow.print(" g"); } }

// Definicija pinova za L298N motor driver
const int motorIn1 = 32;
const int motorIn2 = 33;
const int freq = 30000;
const int motorEnable = 13;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 0;  // Početna vrijednost brzine motora

// Pin za tipku
```

```
const int buttonPin = 12;
const int buttonPin1 = 16;
const int buttonPin2 = 17;
// Promjenjive za praćenje stanja motora
bool motorAllow = true;
bool motorRunning = true;
bool buttonState = HIGH;
bool lastButtonState = HIGH;
unsigned long lastDebounceTime = 0;
bool buttonState1 = HIGH;
bool lastButtonState1 = HIGH;
unsigned long lastDebounceTime1 = 0;
bool buttonState2 = HIGH;
bool lastButtonState2 = HIGH;
unsigned long lastDebounceTime2 = 0;
const unsigned long debounceDelay = 50;

ServoEasing Servo1;
ServoEasing Servo2;
ServoEasing Servo3;

void startMotor(){
  digitalWrite(motorIn1, HIGH);
  digitalWrite(motorIn2, LOW);
  ledcWrite(pwmChannel, 255);
  motorRunning = true; }

void stopMotor(){
  digitalWrite(motorIn1, LOW);
  digitalWrite(motorIn2, LOW);
  ledcWrite(pwmChannel, 0);
```

```
  motorRunning = false;
}
// Funkcija za resetiranje sustava
void systemPaused(){
  motorAllow = false;
  stopMotor();
}
void setup() {
  Serial.begin(9600);


  // Inicijalizacija senzora težine
  scale.begin(DOUT, CLK);
  scale.set_scale(-15314 / 68.5);  // Postavljen broj dobiven kalibracijom vage
  scale.tare();  // Postavi trenutnu težinu kao nulu


  // Inicijalizacija LCD ekrana
  lcdCurrent.init();
  lcdCurrent.backlight();
  lcdCurrent.clear();
  lcdCurrent.setCursor(0, 0);
  lcdCurrent.print("Postavi proizvod"); // ispisuje poruku čim se pokrene program
  lcdRed.init();
  lcdRed.backlight();
  lcdRed.clear();
  lcdRed.setCursor(0, 0);
  lcdRed.print("Crvena: ");     // ispisuje poruku čim se pokrene program
  lcdGreen.init();
  lcdGreen.backlight();
  lcdGreen.clear();
  lcdGreen.setCursor(0, 0);
```

```
lcdGreen.print("Zelena: ");  // ispisuje poruku čim se pokrene program
lcdYellow.init();
lcdYellow.backlight();
lcdYellow.clear();
lcdYellow.setCursor(0, 0);
lcdYellow.print("Zuta: ");    // ispisuje poruku čim se pokrene program

pinMode(motorIn1, OUTPUT); // povezivanje motora sa njegovim driveom kako bi se napajao
pinMode(motorIn2, OUTPUT);
pinMode(motorEnable, OUTPUT);
ledcAttachChannel(motorEnable, pwmChannel, freq, resolution);

pinMode(S0, OUTPUT);   // inicijalizacija pinova senzora boje
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensorOut, INPUT);

// Podesavanje frekvencije skaliranja na 20% senzora boje
digitalWrite(S0, LOW);
digitalWrite(S1, HIGH);

// Inicijalizacija tipki
pinMode(buttonPin, INPUT_PULLUP);  //tipke su aktivne LOW
pinMode(buttonPin1, INPUT_PULLUP);
pinMode(buttonPin2, INPUT_PULLUP);

// Attach servo motori se povezuju na pinove i postavljaju u početni položaj
Servo1.attach(5, 170);
Servo2.attach(19, 38);  // Servo2 inicijalizacija na pin 19
```

```
  Servo3.attach(18, 20);  // Servo3 inicijalizacija na pin 18
}


void loop() {
  int reading = digitalRead(buttonPin);
  if (reading != lastButtonState) {
    lastDebounceTime = millis();
  }
  // koristimo vrijeme od 50ms pritiska tipke kako bi se sprječile slučajne detekcije
  if ((millis() - lastDebounceTime) > debounceDelay) {

    if (reading != buttonState) {
      buttonState = reading;
      if(container.redWeight > 18){              //ako je pun crveni spremnik
        if (buttonState == LOW) {                //ako je pritisnuta crvena tipka
          if (motorRunning == false) {           //ako je motor zaustavljen
            container.redWeight = 0;             //resetiraj masu spremnika
            lcdRed.clear();
            lcdRed.setCursor(0, 0);
            lcdRed.print("Crvena:");
            lcdCurrent.clear();
            lcdCurrent.setCursor(0, 0);
            lcdCurrent.print("Sistem resetiran");          //poruka o resetiranju
            delay(1000);
            lcdCurrent.clear();
            lcdCurrent.setCursor(0, 0);
            lcdCurrent.print("Postavi proizvod");
            motorAllow = true;                           //ponovno pokreni sustav
            startMotor();
          }
        }
```

```
    }   // ukoliko su svi spremnici prazni a tipka pritisnuta, nastaviti sa programom, ovo je slučaj ukoliko se zaustavi traka zbog nedozvoljenog predmeta postavljenog na vagi, pa se program nastavlja pritiskom tipke

    else if(container.yellowWeight <18 && container.redWeight <18 && container.greenWeight <18){
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Postavi proizvod");
      motorAllow = true;
      startMotor();
    }
  }
 }
 lastButtonState = reading;


 int reading1 = digitalRead(buttonPin1); //čitanje pritiska zelene tipke
 if (reading1 != lastButtonState1) {
  lastDebounceTime1 = millis();
 }


 if ((millis() - lastDebounceTime1) > debounceDelay) {


  if (reading1 != buttonState1) {
   buttonState1 = reading1;
   if(container.greenWeight > 18){                  //ako je pun zeleni spremnik
    if (buttonState1 == LOW) {                      //ako je pritisnuta zelena tipka
     if (motorRunning == false) {                   //ako se motor ne kreće
      container.greenWeight = 0;                    //resetiraj zeleni spremnik
```

```
          lcdGreen.clear();
          lcdGreen.setCursor(0, 0);
          lcdGreen.print("Zelena:");
          lcdCurrent.clear();
          lcdCurrent.setCursor(0, 0);
          lcdCurrent.print("Sistem resetiran");
          delay(1000);
          lcdCurrent.clear();
          lcdCurrent.setCursor(0, 0);
          lcdCurrent.print("Postavi proizvod");                    //nastavi sa radom programa
          motorAllow = true;
          startMotor();
        }
      }
    }   // ako spremnici nisu puni a motor je zaustavljen znači da je nedozvoljen proizvod:
   else if(container.yellowWeight <18 && container.redWeight <18 && container.greenWeight
<18){
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Postavi proizvod");      //pa se nakon pritiska tipke resetira program
      motorAllow = true;                        // i nastavlja se motor trake kretati
      startMotor();
    }
  }
 }
 lastButtonState1 = reading1;

 int reading2 = digitalRead(buttonPin2);              // očitavanje žute tipke
 if (reading2 != lastButtonState2) {
```

```
  lastDebounceTime2 = millis();
 }


 if ((millis() - lastDebounceTime2) > debounceDelay) {


  if (reading2 != buttonState2) {
   buttonState2 = reading2;
   if(container.yellowWeight > 18){                //Ako je žuti spremnik pun
    if (buttonState2 == LOW) {                     //ako je pritisnuta žuta tipka
     if (motorRunning == false) {                  //ako se motor ne kreće
      container.yellowWeight = 0;                  //resetiraj žuti spremnik
      lcdYellow.clear();
      lcdYellow.setCursor(0, 0);
      lcdYellow.print("Zuta:");
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Sistem resetiran");
      delay(1000);
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Postavi proizvod");        //program nastavlja s daljnjim radom
      motorAllow = true;
      startMotor();
     }

    }

   }              //ukoliko je motor stao a spremnici nisu puni potrebno je pritisnuti tipku da se program nastavi jer se na vagi pojavio proizvod nepoznate boje zbog koje je motor stao

   else if(container.yellowWeight <18 && container.redWeight <18 && container.greenWeight <18){
    lcdCurrent.clear();
```

```
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Postavi proizvod");
      motorAllow = true;
      startMotor();
    }
  }
}
lastButtonState2 = reading2;


  // Mjerenje težine
delay(1000);
float weight = scale.get_units(10);  // uzima se srednja vrijednost 10 mjerenja
Serial.println(weight);
if(motorAllow == true){
  startMotor();                        // nakon pokretanja programa pokreće se motor pokretne trake

  if (weight > 3.0){                        // ako je masa na vagi veća od 3 grama
    delay(3000);


    // Čitanje crvene frekvencije boje
    digitalWrite(S2, LOW);
    digitalWrite(S3, LOW);
    crvenaFrekvencija = pulseIn(sensorOut, LOW);


    // Čitanje zelene frekvencije boje
    digitalWrite(S2, HIGH);
    digitalWrite(S3, HIGH);
    zelenaFrekvencija = pulseIn(sensorOut, LOW);
```

```
    // Čitanje plave frekvencije boje
    digitalWrite(S2, LOW);
    digitalWrite(S3, HIGH);
    plavaFrekvencija = pulseIn(sensorOut, LOW);


    // Ispis boja u Serial Monitor
    Serial.print("CRVENA = ");
    Serial.println(crvenaFrekvencija);
    Serial.print(" ZELENA = ");
    Serial.println(zelenaFrekvencija);
    Serial.print(" PLAVA = ");
    Serial.println(plavaFrekvencija);
    //ocitanje crvene boje
    if ((crvenaFrekvencija >125 && crvenaFrekvencija <205 && zelenaFrekvencija >275 &&
zelenaFrekvencija <405 &&
        plavaFrekvencija > 225 && plavaFrekvencija < 320) ||
        (crvenaFrekvencija >75 && crvenaFrekvencija <135 && zelenaFrekvencija >165 &&
zelenaFrekvencija <275 &&
        plavaFrekvencija > 125 && plavaFrekvencija < 205)) {
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Crveni: ");
      lcdCurrent.setCursor(0, 1);
      lcdCurrent.print(weight);          //ispisivanje mase i boje na glavni LCD ekran
      lcdCurrent.print(" g");
      delay(3000);


    // Dodavanje težine u crveni spremnik
      container.redWeight += weight;
      checkWeightRed(container.redWeight);


      Servo3.easeTo(20, 200);
```

```
      delay(1000);
//micanje servomotora iznad trake da se usmjeri predmet prema spremniku
      Servo2.easeTo(95, 150);
      Servo1.easeTo(10, 100);  // Aktiviraj servo motor da se pogura predmet na traku
      Servo1.easeTo(170, 100); // vraćanje servomotora kako bi se mogao postaviti novi proizvod
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Postavi proizvod");  // Resetiraj ekran kako bi se mogao novi proizvod
      delay(2000);
      if (container.redWeight >= maxWeight - weightTolerance) {   //ako je spremnik pun
        lcdCurrent.clear();
        lcdCurrent.setCursor(0, 0);
        lcdCurrent.print("Crveni spremnik");
        lcdCurrent.setCursor(0, 1);
        lcdCurrent.print("treba isprazniti");
        delay(2000);
        systemPaused();  // Pauziraj sustav kad je spremnik pun
      }
    }

    // ocitanje zelene boje
    else if ((crvenaFrekvencija > 290 && crvenaFrekvencija <410 && zelenaFrekvencija >245
&& zelenaFrekvencija <305
          && plavaFrekvencija > 235 && plavaFrekvencija < 305) ||
          (crvenaFrekvencija >215 && crvenaFrekvencija <285 && zelenaFrekvencija >145 &&
zelenaFrekvencija <200
           && plavaFrekvencija > 155 && plavaFrekvencija < 200)) {
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Zelena: ");
      lcdCurrent.setCursor(0, 1);
      lcdCurrent.print(weight);
      lcdCurrent.print(" g");              //ispisivanje mase i boje na glavni LCD ekran
```

```
      delay(3000);
      // Dodavanje težine u zeleni spremnik
      container.greenWeight += weight;
      checkWeightGreen(container.greenWeight);


      Servo2.easeTo(38, 200);
      Servo3.easeTo(90, 150);
      delay(1000);
      Servo1.easeTo(10, 100);  // Aktiviraj servo da se pomakne predmet prema pokretnoj traci
      Servo1.easeTo(170, 100);
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
      lcdCurrent.print("Postavi proizvod");  // Resetiraj
      if (container.greenWeight >= maxWeight - weightTolerance) { //ako je pun zeleni spremnik
        lcdCurrent.clear();
        lcdCurrent.setCursor(0, 0);
        lcdCurrent.print("Zeleni spremnik");
        lcdCurrent.setCursor(0, 1);
        lcdCurrent.print("treba isprazniti");
        delay(3000);
        systemPaused();  // Pauziraj sustav kad je spremnik pun
      }
    }
    //provjera zute boje
    else if ((crvenaFrekvencija >85 && crvenaFrekvencija <175 && zelenaFrekvencija >130
&& zelenaFrekvencija <227
          && plavaFrekvencija > 165 && plavaFrekvencija < 250) ||
          (crvenaFrekvencija >55 && crvenaFrekvencija <95 && zelenaFrekvencija >75 &&
zelenaFrekvencija <155
           && plavaFrekvencija > 105 && plavaFrekvencija < 175)) {
      lcdCurrent.clear();
      lcdCurrent.setCursor(0, 0);
```

```
  lcdCurrent.print("Zuta: ");
  lcdCurrent.setCursor(0, 1);
  lcdCurrent.print(weight);
  lcdCurrent.print(" g");              //ispis na glavni ekran boja i masa postavljenog predmeta
  delay(3000);

  // Dodavanje težine u zeleni spremnik
  container.yellowWeight += weight;
  checkWeightYellow(container.yellowWeight);

  Servo2.easeTo(38, 200);
  delay(1000);
  Servo3.easeTo(20, 150);
  Servo1.easeTo(10, 100);  // Aktiviraj
  Servo1.easeTo(170, 100);  // Resetiraj
  lcdCurrent.clear();
  lcdCurrent.setCursor(0, 0);
  lcdCurrent.print("Postavi proizvod");
  if (container.yellowWeight >= maxWeight - weightTolerance) {
    lcdCurrent.clear();
    lcdCurrent.setCursor(0, 0);
    lcdCurrent.print("Zuti spremnik");
    lcdCurrent.setCursor(0, 1);
    lcdCurrent.print("treba isprazniti");
    delay(5000);
    systemPaused();  // Pauziraj sustav kad je spremnik pun
  }
}
else {          //ukoliko nije prepoznata ni crvena ni žuta ni zelena boja onda je nepoznata
  lcdCurrent.clear();
  lcdCurrent.setCursor(0, 0);
```

```
      lcdCurrent.print("Postavi proizvod");
      lcdCurrent.setCursor(0, 1);
      lcdCurrent.print("ispravne boje");
      systemPaused();                          // pauziraj sve kada je nedozvoljena boja
    }
  }
 }
}
```