

Modeliranje sustava vođenja automatske akvarijske hranilice

Martić, Karlo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:797688>

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

Download date / Datum preuzimanja: **2025-04-02**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Karlo Martić

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

doc. dr. sc. Mihael Cipek

Student:

Karlo Martić

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se svojoj obitelji, prijateljima i Ivanu Janđelu na potpori, te svom mentoru doc. dr.sc. Mihaelu Cipeku na podršci, savjetima i pomoći prilikom pisanja završnog rada.

Karlo Martić



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 24 – 06 / 1	
Ur.broj: 15 – 24 –	

ZAVRŠNI ZADATAK

Student: **Karlo Martić** JMBAG: **0035239853**

Naslov rada na hrvatskom jeziku: **Modeliranje sustava vođenja automatske akvarijske hranilice**

Naslov rada na engleskom jeziku: **Modelling of the automatic aquarium feeder control system**

Opis zadatka:

Akvarij je prostor ili posuda ispunjena vodom s prozirnom plohom u kojoj se čuvaju vodene životinje. Uobičajeno se uz akvarij nalazi i specijalizirana oprema za održavanje kvalitete vode, topline te ostalih karakterističnih uvjeta za život jedinki u akvariju. Jedan takav sustav specijalizirane opreme je automatska hranilica za akvarij koja se može programirati kako bi omogućila adekvatnu količinu hrane u odgovarajućim vremenskim intervalima te također po potrebi korisniku javljala određena stanja. Ovaj rad se bavi modeliranjem automatskog sustava za hranjenje riba u akvariju. U zadatku je potrebno:

- opisati problem i načine hranjenja riba u različitim akvarijskim sustavima
- izraditi računalni model jednog odabranog primjera hranilice
- predložiti sustav vođenja hranilice
- provesti simulaciju rada sustava
- na temelju rezultata dati zaključak i eventualne smjernice za poboljšanja.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

24. 4. 2024.

Datum predaje rada:

2. rok (izvanredni): 11. 7. 2024.
3. rok: 19. i 20. 9. 2024.

Predvideni datumi obrane:

2. rok (izvanredni): 15. 7. 2024.
3. rok: 23. 9. – 27. 9. 2024.

Zadatak zadao:


doc. dr. sc. Mihael Cipek

Predsjednik Povjerenstva:


izv. prof. dr. sc. Petar Čurković

SADRŽAJ

SADRŽAJ	1
POPIS SLIKA.....	3
POPIS TABLICA.....	5
SAŽETAK.....	6
SUMMARY	7
1. UVOD.....	8
2. PREGLED UVJETA KOJE HRANILICA TREBA ZADOVOLJITI	9
2.1. Vrste akvarijskih riba i njihove potrebe	9
2.1.1. Zlatna ribica (goldfish, <i>Carassius auratus</i>).....	9
2.1.2. Anđeoska ribica (skalar, angelfish, <i>Pterophyllum scalare</i>).....	10
2.2. Hrana.....	11
2.3. Akvariji	11
3. PRIJEDNOG RJEŠENJA REALIZACIJE AUTOMATIZIRANE HRANILICE ZA RIBE..	13
3.1. Prijedlog konstrukcijskog rješenja.....	14
3.2. Opis komponenata elektroničkog spoja	16
3.2.1. Arduino Nano.....	17
3.2.2. Ultrazvučni senzor HC-SR04	18
3.2.3. LCD1602 zaslon s I2C sučeljem.....	20
3.2.4. Bluetooth modul HC-06.....	21
3.2.5. Mini DC motor.....	23
3.2.6. 5 – segmentni LED zaslon / kontrolna lampica	24
3.2.7. Tipkala / tipkovnički modul	25
3.2.8. Baterija 9V	26
3.2.9. Žice i konektor za bateriju	27
3.2.10. L293D motor kontroler	28
3.2.11. 8-bitni shift registar.....	30
4. PROGRAMIRANJE.....	32
4.1. Detaljan opis rada	32
4.2. Arduino kod	34
4.2.1. Deklaracija varijabli 1. dio – definiranje pinova i biblioteke	34
4.2.2. Deklaracija varijabli 2. dio – definiranje vrijednosti	35

4.2.3.	Inicijalizacijska rutina (void setup).....	36
4.2.4.	Petlja (void loop) 1. dio – Kontrola količine hrane	37
4.2.5.	Petlja (void loop) 2. dio – Upravljanje pomoću tipkala.....	38
4.2.6.	Petlja (void loop) 3. dio – Upravljanje pomoću aplikacije preko Bluetooth modula.....	39
4.2.7.	Petlja (void loop) 4. dio – Kontrola pozadinskog osvjetljenja LCD zaslona..	40
4.2.8.	Petlja (void loop) 5. dio – Kontrola rada DC motora	40
4.2.9.	Petlja (void loop) 6. dio – Slanje podataka na aplikaciju preko Bluetooth modula.....	41
4.2.10.	Ispisivanje podataka na LCD zaslonu.....	41
4.3.	MIT App Inventer kod	42
4.3.1.	Povezivanje s Arduinoom preko Bluetooth modula	42
4.3.2.	Definiranje tipkala za vrijeme i količinu	43
4.3.3.	Definiranje + i – tipkala	44
4.3.4.	Primanje podataka s Arduina o vremenu, količini i stanju količine hrane.....	45
4.3.5.	Dizajn aplikacije	46
5.	SPAJANJE I TESTIRANJE (SIMULACIJA).....	48
5.1.	Električna shema.....	48
5.2.	Spajanje i testiranje (simulacija).....	50
5.3.	Diskusija	51
6.	ZAKLJUČAK.....	53
	LITERATURA.....	54

POPIS SLIKA

Slika 2.1. Zlatna ribica	10
Slika 2.2. Anđeoska ribica	10
Slika 2.3. Hrana za ribe.....	11
Slika 2.4. Akvarij zapremnine 200l	12
Slika 3.1. Funkcijski blok dijagram ulaznih i izlaznih varijabli Arduino mikrokontrolera	13
Slika 3.2. Računalni 3D model sklopa.....	15
Slika 3.3. Fotografija elektroničkog sklopa	16
Slika 3.4. Arduino Nano i njegov položaj u sklopu	17
Slika 3.5. Ultrazvučni senzor i njegov položaj u sklopu	19
Slika 3.6. LCD zaslon i njegov položaj u sklopu	20
Slika 3.7. Bluetooth modul HC-06 i njegov položaj u sklopu	22
Slika 3.8. Mini DC motor i njegov položaj u sklopu	23
Slika 3.9. 5 – segmentni LED zaslon i njegov položaj u sklopu	25
Slika 3.10. Tipkala i njihov položaj u sklopu	26
Slika 3.11. Baterija od 9V i njezin položaj u sklopu	27
Slika 3.12. Konektor za bateriju od 9V	28
Slika 3.13. L293D motor kontroler i njegov položaj u sklopu	29
Slika 3.14. 8-bitni shift registar i njegov položaj u sklopu	31
Slika 4.1. Deklaracija varijabli 1. dio – definiranje pinova i biblioteke	34
Slika 4.2. Deklaracija varijabli 2. dio – definiranje vrijednosti.....	35
Slika 4.3. Inicijalizacijska rutina (void setup)	36
Slika 4.4. Petlja (void loop) 1. dio – Kontrola količine hrane	37
Slika 4.5. Petlja (void loop) 2. dio – Upravljanje pomoću tipkala.....	38
Slika 4.6. Petlja (void loop) 3. dio – Upravljanje pomoću aplikacije preko Bluetooth modula	39
Slika 4.7. Petlja (void loop) 4. dio – Kontrola pozadinskog osvjetljenja LCD zaslona	40
Slika 4.8. Petlja (void loop) 5. dio – Kontrola rada DC motora	40
Slika 4.9. Petlja (void loop) 6. dio – Slanje podataka na aplikaciju preko Bluetooth modula.....	41
Slika 4.10. Ispisivanje podataka na LCD zaslonu.....	41
Slika 4.11. Izgled ikone aplikacija na početnom zaslonu mobilnog telefona	42
Slika 4.12. Povezivanje s Arduinoom preko Bluetooth modula	42

Slika 4.13. Definiranje tipkala za vrijeme i količinu	43
Slika 4.14. Definiranje + i – tipkala	44
Slika 4.15. Primanje podataka s Arduina o vremenu, količini i stanju količine hrane	45
Slika 4.16. Izgled dizajna aplikacije u programu MIT App Inventor	46
Slika 4.17. Izgled gotove aplikacije koja nije povezana s Arduinom	47
Slika 5.1. Električna shema.....	49
Slika 5.2. Eksperimentalni spoj	50
Slika 5.3. Primjer promjene boje prikaza promjenom stanja u spremniku hranilice	51

POPIS TABLICA

Tablica 3.1. Specifikacije Arduino Nano razvojne pločice	18
Tablica 3.2. Specifikacije HC-SR04 ultrazvučnog senzora	19
Tablica 3.3. Specifikacije LCD1602 zaslona	21
Tablica 3.4. Specifikacije Bluetooth modula HC-06	22
Tablica 3.5. Specifikacije Mini DC motora	24
Tablica 3.6. Specifikacije 5 – segmentnog LED zaslona	25
Tablica 3.7. Specifikacije tipkala	26
Tablica 3.8. Specifikacije baterije od 9V	27
Tablica 3.9. Specifikacije izoliranih žica promjera 1,36mm	28
Tablica 3.10. Specifikacije L293D motor kontrolera.....	29
Tablica 3.11. Specifikacije 8-bitni shift registra	31

SAŽETAK

U ovom radu razrađen je i riješen problem hranjenja akvarijskih riba pomoću automatske akvarijske hranilice. U početku rada dan je opis i dnevne potrebe za hranom uobičajenih akvarijskih riba, kao i kratak pregled akvarija na tržištu. Potom je predloženo konstrukcijsko rješenje automatizirane hranilice, te je prikazan njen računalni model. Također, u radu je prikazana elektronička shema kao i simulacija s stvarnim komponentama sustava. Priložen je i primjer koda kojim se vrši upravljanje hranilicom, kao i mobilna aplikacija za lakšu interakciju s istom. Računalni model i tehnička dokumentacija izrađeni su u programu SolidWorks 2020, kodiranje je napravljeno u programu Arduino IDE, mobilna aplikacija izrađena pomoću web stanice MIT App Inventor, dok je električna shema izrađena pomoću web stranice Circuit Diagram Web Editor.

Ključne riječi: Hranilica za ribe, mobilna aplikacija, Arduino, modeliranje, simulacija

SUMMARY

Problem of feeding aquarium fishes using an automatic aquarium feeder has been elaborated and solved in this undergraduate thesis. The first part of the thesis is a description of the daily food needs of the most common aquarium fish, as well as a brief overview of aquariums on the market. In the second part of the thesis, the construction solution of the automatic feeder was proposed, and its computer model was presented. Also, in the thesis was presented an electronic scheme as well as a simulation with real system components. An example of the code used to control the feeder is also presented, as well as a mobile application for easy interaction with it. The computer model and technical documentation were created in the SolidWorks 2020 program, the coding was done in the Arduino IDE program, the mobile application was created with the MIT App Inventor website, while the electrical scheme was created with the Circuit Diagram Web Editor.

Keywords: Fish feeder, mobile application, Arduino, modeling, simulation

1. UVOD

Prvi zapisi o ribama u zatočeništvu u ljudskim nastambama sežu još iz doba starog Sumera od prije 4500 godina kada su se manje vrste riba počele držati u glinenim posudama, no tada je primarna svrha tog čina bila osiguravanje alternativnog izvora svježe riblje hrane budući da je to bio jedini način očuvanja svježe ribe za konzumaciju nakon što se riba ulovila iz svoga staništa. Prvi narod koji je držao ribe kao kućne ljubimce bili su drevni Kinezi koji su malene egzotične vrste riba poput (i danas najpopularnije vrste ribe – kućnog ljubimca) zlatne ribice čuvali u keramičkim posudama i vazama. Isto tako, postoje iskopine iz doba Rimskog carstva koje potvrđuju postojanje bazena – akvarija s murinama koje su, isto kao i u doba drevnih Sumerana, prvenstveno služile za hranu. Prvi stakleni akvarij pojavio se u Engleskoj 1700-tih u obliku staklene vaze dok se prvi akvarij današnjeg tipa pojavio u Njemačkoj 1800-tih. Prvi zapisi o akvariju današnjeg tipa nalaze se u knjizi Philipa Henryaa Gossea iz 1854-e (*The Aquarium: An Unceiling of the Wonders of the Deep Sea*) [1].

Glavni problem držanja riba u akvariju jest, uz redovitu zamjenu vode i čišćenje istog, hranjenje ribe. Primarni izvor hrane za malene ribe jesu planktoni i alge dok su za veće ribe to manji stanovnici voda i insekti. U akvarijskom okruženju ribe nemaju mogućnost „prirodne“ potrage za hranom zbog ograničenog prostora umjetno napravljenog staništa (akvarija) u kojem je voda najčešće potpuno lišena planktona, algi te drugih sitnijih riba ili rakova. Stoga, kako bi preživjele, ovise o čovjekovoj opskrbi hranom. Budući da je u današnjem užurbanom okruženju čovjek zaokupljen raznim obvezama, često se dogodi da nenamjerno zaboravi nahraniti ribe ili im u žurbi da preveliku/premalenu količinu hrane. Zbog tog čovjekovog slučajnog nemara riba ne dobiva adekvatne uvjete za preživljavanje što rezultira njenom bolešću ili ugibanjem. Kako bi se spriječio takav scenarij, u ovom radu razradio se koncept automatizirane programibilne hranilice za ribe koja, u slučaju malog broja malih riba, može raditi i hraniti ribe bez uplitanja čovjeka u njezin rad i po više tjedana u komadu. U ovom radu prvo je dan kratki pregled akvarija i najčešćih vrsta riba koje se drže u njima, te potom i cjelovito rješenje problema.

2. PREGLED UVJETA KOJE HRANILICA TREBA ZADOVOLJITI

Hranilica za ribe razmatrana u ovom radu prvenstveno je namijenjena za malene ribe koje su najčešće prisute kao kućni ljubimci u akvarijima unutar stanova, ureda i poslovnih prostora zbog njihovih zahtjeva za relativno malim akvarijima te zbog lagane brige o njima (iako se razmjernom skalacijom elemenata hranilice ona može prilagoditi za gotovo sve vrste akvarijskih riba).

2.1. Vrste akvarijskih riba i njihove potrebe

Najčešće vrste akvarijskih ribica su: plava neunka, gupi, sijamski borac, dugina ribica, te zlatna i anđeoska ribica. Ovaj rad fokusira se na potonje dvije vrste budući da su one najprisutnije u akvarijima, te su također najvećih dimenzija između navedenih vrsta ribica te tako shodno tome imaju najveće potrebe (konzumiraju najviše hrane i zahtijevaju najveći akvarij) [2].

2.1.1. Zlatna ribica (*goldfish*, *Carassius auratus*)

Zlatna ribica (*Carassius auratus*) pripada porodici šarana i danas je najzastupljenija vrsta u akvarijima. Ona je jedna od prvih riba koja je "udomaćena". U prirodnim uvjetima može narasti do 50-60 cm dužine i doseći težinu do 3 kg. U akvarijima najčešće dostižu maksimalnu dužinu od oko 15 cm. Radi društva preporučljivo je u akvariju držati najmanje 3 zlatne ribice istovremeno iako je prosjek 5 zlatnih ribica. Za svaki centimetar ribice potrebno je osigurati 2 litre vode u akvariju, što dovodi do veličine akvarija volumena 150 litara. Budući da ribica pojede svu hranu koja joj se ponudi, to u konačnici dovodi do prejedanja ribice te do njenog ugibanja. Ribica bi trebala 1-2 puta dnevno dobiti količinu hrane koju može pojesti u 2 minute, što iznosi oko 0,4 grama dnevno (oko 2% ribičije težine koja iznosi oko 20 grama). To dovodi do dnevne konzumacije hrane u iznosu 2 grama za 5 ribica (ili 2x1g dnevno) [3, 4].



Slika 2.1. Zlatna ribica [5]

2.1.2. *Anđeoska ribica (skalar, angelfish, Pterophyllum scalare)*

Skalar (*Pterophyllum scalare*) je popularna akvarijska riba iz porodice ciklida, porijeklom iz slatkovodnih rijeka Južne Amerike. U prirodnim uvjetima može narasti i do 25 cm u dužinu i 20 cm u visinu. U akvarijima obično dostiže nešto manje dimenzije, ali i dalje ostaje impresivna svojim izgledom i ponašanjem. Budući da su to vrste ribica koje žive u paru, poželjno je u istom akvariju imati 2-3 para ribica (6 riba). Volumen akvarija kojeg je potrebno imati za adekvatan život 2-3 para ribica je 200 litara. Skalar bi trebao 1-2 dnevno dobiti količinu hrane koja odgovara 3% njegove mase koja iznosi 15-20 grama. To dovodi do dnevne konzumacije hrane u iznosu 3,6 grama za 6 ribica (ili 2x1,8g dnevno) [6, 7].



Slika 2.2. Anđeoska ribica [8]

2.2. Hrana

Iako se ribice u svom prirodnom okolišu hrane raznim vrstama prirodno dostupne hrane (sitni insekti, larve, biljna materija, planktoni te crvi i mali rakovi), u akvarijskom okruženju ograničene su na tvornički prerađenu hranu u listićima koja se sastoji od žitarica, riba i prerađevina od ribe, soje, kvasca, račića, algi i aloevera. Hrana se najčešće prodaje u pakiranjima od 100 ml, 250 ml ili 1200 ml. 100 ml hrane teži oko 25 g, što dovodi do izračuna da za jedan prosječni akvarij s homogenim vrstama ribe 100 ml hrane traje oko 7 dana. Veličina listića je oko 1,5 – 2 mm, dok je njihova debljina puno manja od 1mm [9].



Slika 2.3. Hrana za ribe [9]

2.3. Akvariji

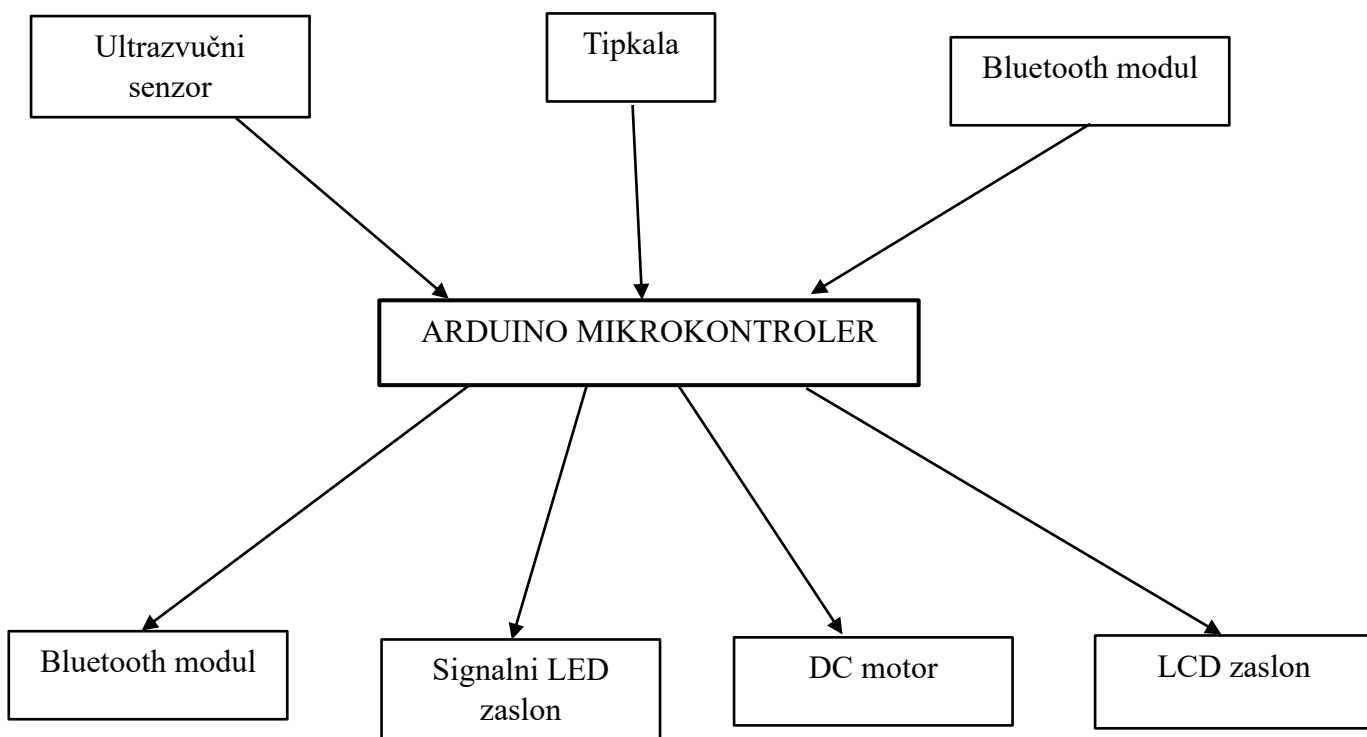
Postoje različiti oblici akvarija, dok su oni najčešći okrugli akvariji (u obliku posude) te oni u obliku kocke ili pravokutnika. Materijal od kojeg se izrađuju akvarij je nepromijenjen već zadnjih 300 godina, a to je staklo. Taj materijal je izabran iz jednostavnog razloga jer njegova prozirnost nudi pogled na dnevne aktivnosti kućnog ljubimca i lakše primjećivanje eventualnih nepravilnosti u njihovom ponašanju, a isto tako pruža ribici izvor dnevne svjetlosti koja je nužna za njen život (za tropske ribice minimalni potrebni izvor sunčeve ili umjetne svjetlosti je oko 12 sati dnevno). Dimenzije akvarija od 200 litara su obično 100 cm x 50 cm x 40 cm. Debljine stakla iznose od 4 mm do 8 mm (ovisno o zapremnini akvarija).



Slika 2.4. Akvarij zapremnine 200l [10]

3. PRIJEDNOG RJEŠENJA REALIZACIJE AUTOMATIZIRANE HRANILICE ZA RIBE

Princip rada temeljen je na sustavu upravljanim Arduino mikrokontrolerom. Sustav naredbe korisnika prima preko fizičkih tipkala pozicioniranih na glavnoj nosivoj konstrukciji hranilice te korisniku prikazuje odabrana stanja na LCD zaslonu (koji se nalazi na kućištu hranilice). Alternativni način upravljanja je preko mobilne aplikacije koja se s Arduino mikrokontrolerom povezuje putem Bluetooth modula. Na temelju namještenih parametara Arduino mikrokontroler kontrolira vrijeme uključivanja i isključivanja DC motora koji pokreće dobavu hrane u akvarij. Preko ultrazvučnog senzora mjeri se količina hrane u spremniku hranilice te se podaci šalju Arduino mikrokontroleru koji na temelju istih kontrolira stanje signalnog LED zaslona te na taj način komunicira s korisnikom.

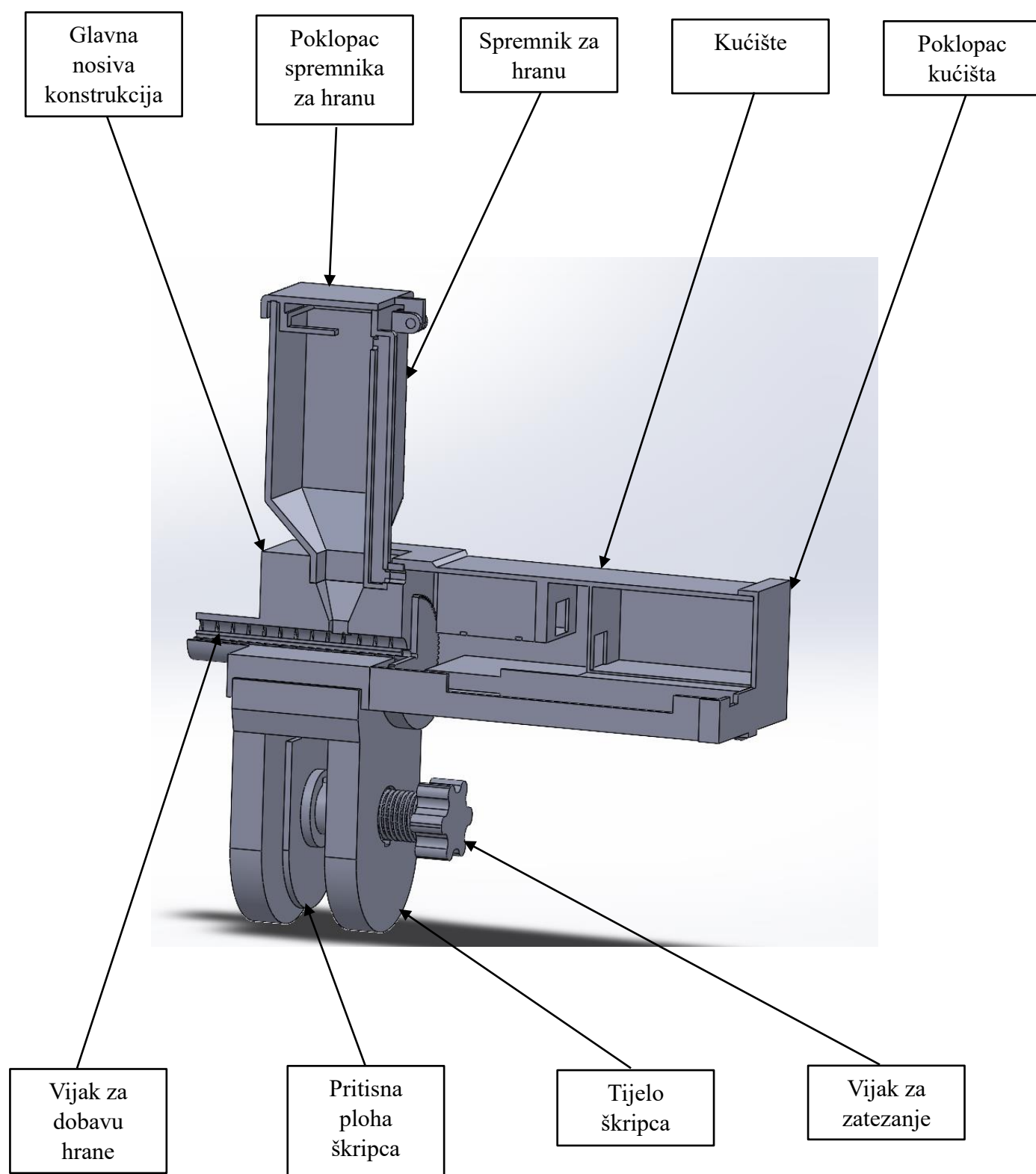


Slika 3.1. Funkcijski blok dijagram ulaznih i izlaznih varijabli Arduino mikrokontrolera

3.1. Prijedlog konstrukcijskog rješenja

Tijelo hranilice se sastoji od 9 dijelova koji su međusobno kompatibilni i zajedno kao cjelina čine hranilicu za ribe. Način spajanja dijelova su umetanje u predviđene utore za specifičan dio (sa ili bez lijepljenja dijelova) te spajanje vijcima. Sklop je izrađen na taj način da se cijela hranilica može rastaviti radi održavanja ili otklona kvara. Predviđena tehnologija izrade je 3D printanje. U ovom poglavlju se objašnjava namjena svakog od tih dijelova te kako funkcioniraju kao cjelina.

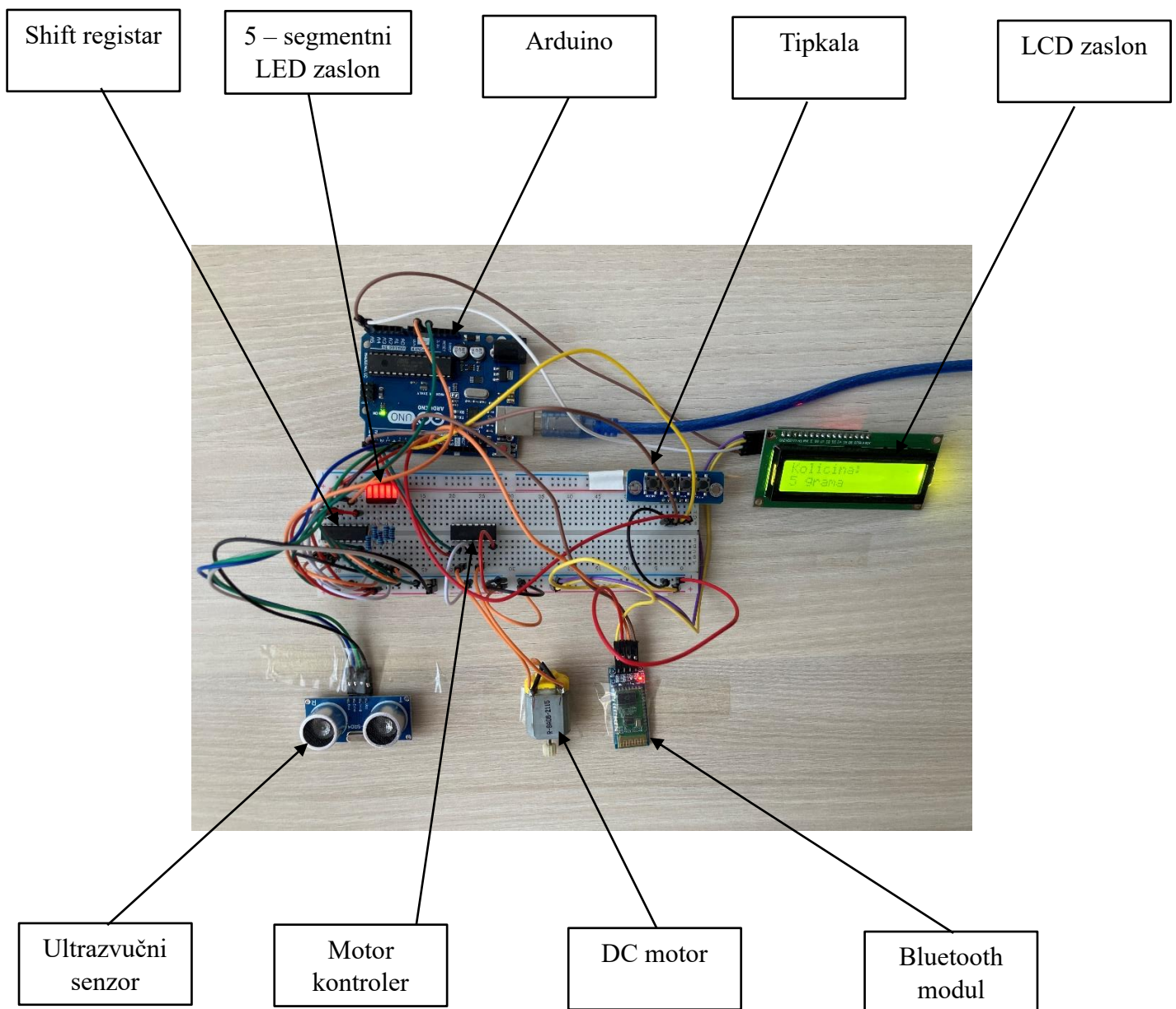
- a) Glavna nosiva konstrukcija – baza sklopa s obzirom da se na nju spajaju spremnik za hranu, kućište, tijelo škripca i vijak za dobavu. U njenoj unutrašnjosti nalazi se lijevak i kanal za dobavu preko kojih hrana od spremnika dolazi do akvarija. Također, u njoj su smještena tipkala te ima kanalice za prolaz žica komponenta do kućišta.
- b) Spremnik za hranu – služi za skladištenje hrane. Povezuje se umetanjem na glavnu nosivu konstrukciju. Na vanjskoj oplati ima kanalicu za žice ultrazvučnog senzora.
- c) Poklopac spremnika za hranu – primarna svrha mu je sprječavanje kontaminacije hrane vanjskim nečistoćama. Na njemu se nalazi ultrazvučnog senzor. Povezan je šarkama s spremnikom za hranu.
- d) Kućište – u njemu je smještena većina elektroničkih komponenti. Povezuje se s glavnom nosivom konstrukcijom pomoću vijaka.
- e) Poklopac kućišta – osim kao nosač prekidača i LED zaslona, služi i kao lako pristupni otvor za mijenjanje baterije. Način spajanja na kućište jest umetanje i vijak.
- f) Vijak za dobavu hrane – pogonjen DC motorom preko zupčanika, provodi hranu od lijevka glavne nosive konstrukcije (iz spremnika za hranu) do akvarija.
- g) Tijelo škripca – svrha mu je pozicioniranje i pričvršćivanje hranilice na staklo akvarija. Spojen je s glavnom nosivom konstrukcijom umetanjem.
- h) Pritisna ploča škripca – služi za jednoličnije raspoređivanje sile pritiska od vijka za zatezanje na staklo. Spaja se na tijelo škripca umetanjem.
- i) Vijak za zatezanje – pomoću njega se postiže pritisak škripca na staklo i učvršćivanje hranilice na akvarij.



Slika 3.2. Računalni 3D model sklopa

3.2. Opis komponenata elektroničkog spoja

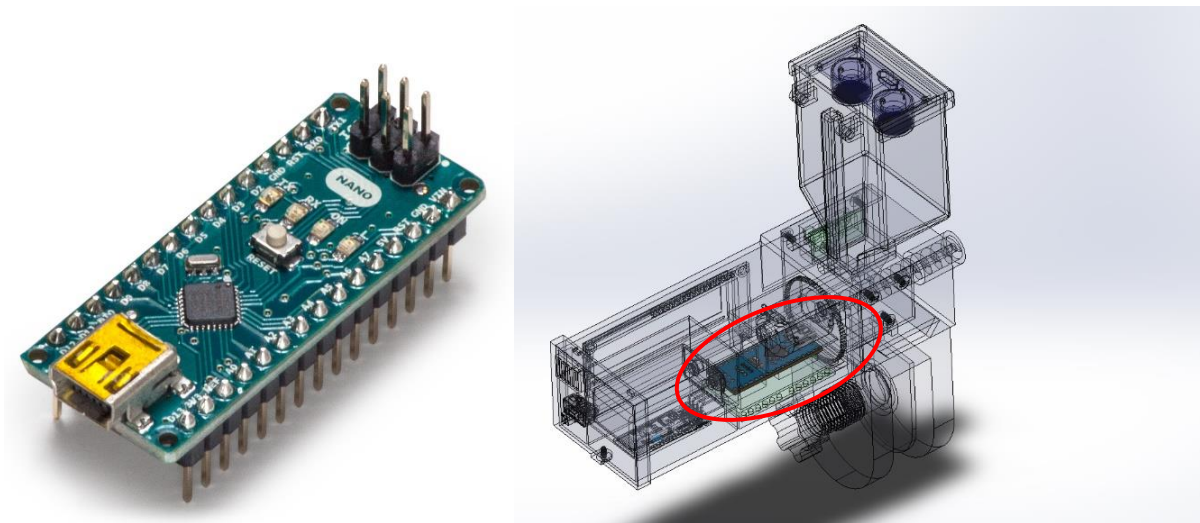
Kako bi se automatizirana hranilica mogla realizirati, potrebno je odabrati komponente koje omogućavaju njen nesmetani i pouzdani rad. Odabir komponenata baziran je na njihovoj cjenovnoj isplativosti i dostupnosti kao i međusobnoj kompatibilnosti. Sve prikazane komponente su komponente elektroničkog sustava (nema mehaničkih, pneumatskim i sličnih komponenti).



Slika 3.3. Fotografija elektroničkog sklopa

3.2.1. Arduino Nano

Arduino Nano je kompaktna razvojna pločica koja se temelji na mikrokontroleru ATmega328P. Iako su njene dimenzijama samo 45 x 18 mm, Arduino Nano nudi sve funkcionalnosti veće Arduino Uno pločice, ali u smanjenom formatu. Opremljen je s 14 digitalnih ulazno/izlaznih pinova od kojih šest može služiti kao PWM izlazi te s osam analognih ulaza što omogućuje široku primjenu u senzoricima, robotici i slično. Povezivanje s računalom omogućuje USB C ili USB mini port. Radi na radnom naponu od 5V i podržava ulazni napon između 7-12V (putem Vin pina) čime je omogućeno napajanje s vanjskog, baterijskog izvora. Arduino Nano dolazi s ugrađenim reset tipkalom i LED indikatorom što olakšava testiranje i rješavanje mogućih poteškoća u razvoju. Budući da podržava sve standardne Arduino biblioteke (library-je), omogućava jednostavan razvoj i integraciju u veće projekte. Također, bitna prednost je njegova pristupačna cijena (2 -27 dolara, ovisi o proizvođaču) [11].



Slika 3.4. Arduino Nano i njegov položaj u sklopu [12]

Tablica 3.1. Specifikacije Arduino Nano razvojne pločice

Mikrokontroler	ATmega328	DC struja po I/O pinu	20 mA (I/O Pins)
Arhitektura	AVR	Ulazni napon	7-12V
Radni napon	5 V	Digitalni I/O pinovi	22 (6 of which are PWM)
Flash memorija	32 KB of which 2 KB used by bootloader	PWM izlaz	6
SRAM	2 KB	Potrošnja struje	19 mA
Brzina osvježavanja	16 MHz	PCB veličina	18 x 45 mm
Analogni ulazi	8	Masa	7 g
EEPROM	1 KB	Kod proizvoda	A000005

3.2.2. Ultrazvučni senzor HC-SR04

Ultrazvučni senzor HC-SR04 je senzor koji se koristi za mjerenje udaljenosti pomoću ultrazvučnih valova. Ovaj senzor se često primjenjuje u raznim projektima gdje je potrebno precizno mjerenje udaljenosti. Princip rada HC-SR04 temelji se na emitiranju ultrazvučnog signala putem svog "trig" pina koji se odbija od objekta i vraća natrag prema senzoru. Kada signal dopije nazad do senzora, on se prima preko "echo" pina. Senzor tada mjeri vrijeme koje je prošlo između emitiranja i primanja signala koristeći brzinu zvuka u zraku (približno 343 metra u sekundi) za izračunavanje udaljenosti objekta. HC-SR04 može mjeriti udaljenosti od 2 cm do 400 cm s točnošću od oko 3 mm. Senzor je kompatibilan s mikrokontrolerima kao što su Arduino, Raspberry Pi i slično. Lako se integrira zahvaljujući svojoj jednostavnosti, pouzdanosti i dostupnosti zbog male nabavne cijene (manje od 2 dolara) [13].

Primjena ultrazvučnog senzora kod hranilice jest mjerenje količine hrane u spremniku.

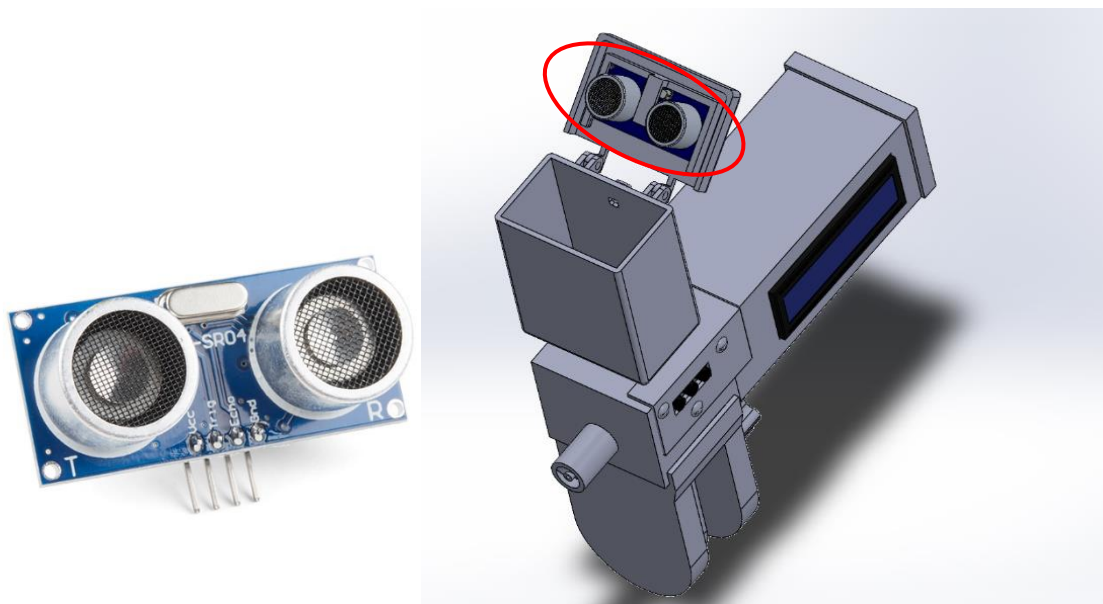
Spajanje ultrazvučnog senzora izvršeno je na slijedeći način:

VCC pin – napajanje od 5V na Arduinu

GND pin – GND pin na Arduinu (-)

TRIG pin – pin 6 na digitalnom ulazu Arduina

ECHO pin – pin 5 na digitalnom ulazu Arduina



Slika 3.5. Ultrazvučni senzor i njegov položaj u sklopu [13]

Tablica 3.2. Specifikacije HC-SR04 ultrazvučnog senzora

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

3.2.3. LCD1602 zaslon s I2C sučeljem

LCD zaslon koristi se u raznim elektroničkim projektima, a posebno je pogodan zbog svoje jednostavnosti prilikom spajanja na mikrokontroler. Ovaj zaslon ima 16 znakova u 2 reda što znači da može prikazati ukupno 32 znaka. Ključna prednost LCD1602 I2C zaslona u odnosu na klasične LCD zaslone je integrirano I2C sučelje koji znatno pojednostavljuje povezivanje s mikrokontrolerom. Umjesto korištenja više pinova (obično 6 ili 8) za upravljanje zaslonom, I2C verzija koristi samo dva pina za komunikaciju i dva za napajanje što ostavlja više slobodnih pinova na mikrokontroleru za druge funkcije. I2C sučelje također omogućuje serijsku komunikaciju mikrokontrolera s LCD zaslonom, a upravljanje je moguće pomoću specijaliziranih biblioteka koje pojednostavljuju kodiranje. Jačinu pozadinskog svjetla moguće je podesiti putem potenciometra na poleđini I2C sučelja. Također, zbog svoje pouzdanosti i dostupnosti koju karakterizira mala nabavna cijena (oko 3 dolara), poželjan je izbor kod Arduino projekata [14].

Primjena LCD zaslona s I2C sučeljem kod hranilice jest prikaz parametara hranilice korisniku.

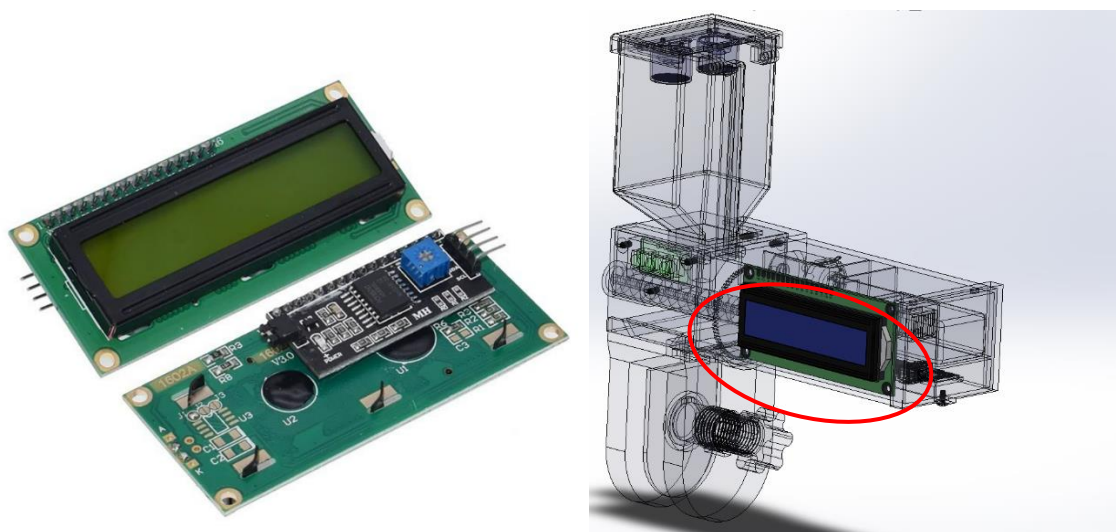
Spajanje LCD zaslona s I2C sučeljem izvršeno je na slijedeći način:

VCC pin – napajanje od 5V na Arduinu

GND pin – GND pin na Arduinu (-)

SDA pin – pin A4 na analognom ulazu Arduina

SCL pin – pin A5 na analognom ulazu Arduina



Slika 3.6. LCD zaslon i njegov položaj u sklopu [15]

Tablica 3.3. Specifikacije LCD1602 zaslona

Radni napon	3.3V/5V	Radna struja	26mA(5V), 13mA (3.3V)
Komunikacijsko sučelje	I2C	VCC	3.3V/5V radni napon
Vrsta zaslona	LCD zaslon	GND	Ground
Kontrolni procesor	AiP31068L	SCL	I2C clock cable
Dimenzije zaslona	64.5 x 16.0 mm	SDA	I2C data cable
Vanjske dimenzije	87.0 x 32.0 x 13.0 mm		

3.2.4. Bluetooth modul HC-06

Bluetooth modul HC-06 je modul koji se koristi za bežičnu serijsku komunikaciju između mikrokontrolera i drugih Bluetooth uređaja. HC-06 radi u načinu rada „slave“ što znači da se može povezati samo s glavnim uređajem (npr. pametnim telefonom ili računalom), ali ne može inicirati vezu samostalno. Modul koristi serijsku komunikaciju (UART) s mikrokontrolerom što omogućuje slanje i primanje podataka putem Bluetootha. Podaci se prenose brzinom od 9600 bita po sekundi. Povezivanje HC-06 modula s mikrokontrolerom je vrlo jednostavno jer se koriste samo 4 žice: VCC (napajanje), GND, TXD (transmit) i RXD (receive). Modul omogućuje bežičnu komunikaciju na udaljenosti do 10 metara što je dovoljno za većinu aplikacija koje se koriste u manjim prostorima, poput kuća i stanova. Zbog svoje dostupnosti (oko 3 dolara) i jednostavne integracije s mikrokontrolerom, primarni je izbor u većini primjena koje zahtijevaju bluetooth komunikaciju [16].

Primjena Bluetooth modula kod hranilice jest pružanje mogućnosti upravljanja hranilicom pomoću aplikacije na mobilnom telefonu.

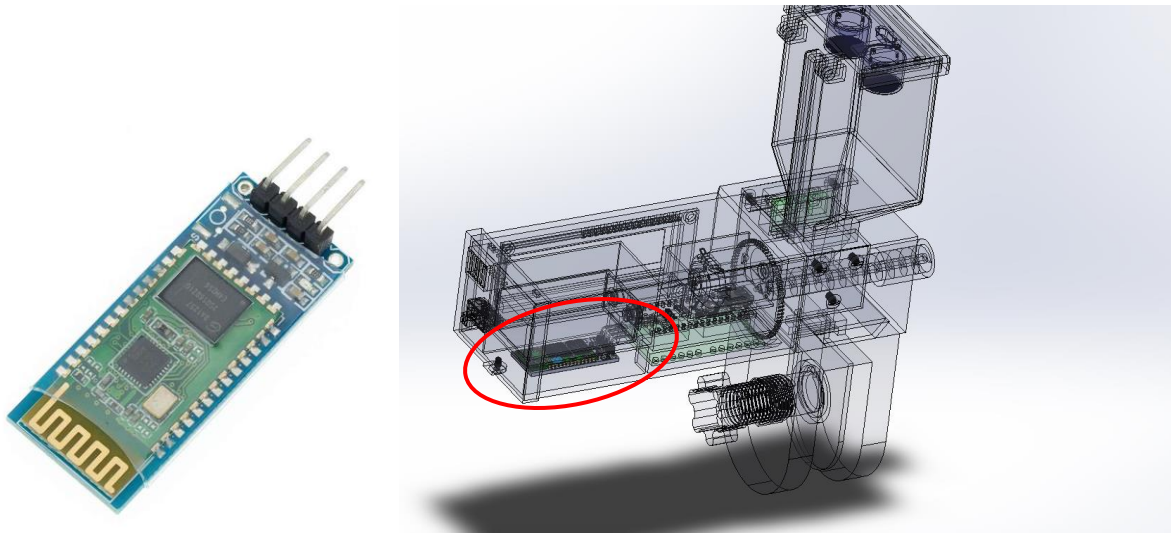
Spajanje Bluetooth modula izvršeno je na slijedeći način:

VCC pin – napajanje od 5V na Arduinu

GND pin – GND pin na Arduinu (-)

RXD pin – pin 10 na digitalnom ulazu Arduina

TXD pin – pin 11 na digitalnom ulazu Arduina



Slika 3.7. Bluetooth modul HC-06 i njegov položaj u sklopu [17]

Tablica 3.4. Specifikacije Bluetooth modula HC-06

Bluetooth protokol	Bluetooth 2.0+EDR (Enhanced Data Rate)	Sinkrona brzina	1 MBps/1 MBps
Komunikacijska frekvencija	2.4GHz	Sigurnost	Identity Validation and Encryption
Preciznost	≤ -80 dBm	Radni napon	3.6-5V (3.6V preporučeno)
Izlazna snaga	$\leq +4$ dBm	Struja	50 mA
Asinkrona brzina	2.1 MBps/160 KBps	Dimenzije	43x16x7mm

3.2.5. Mini DC motor

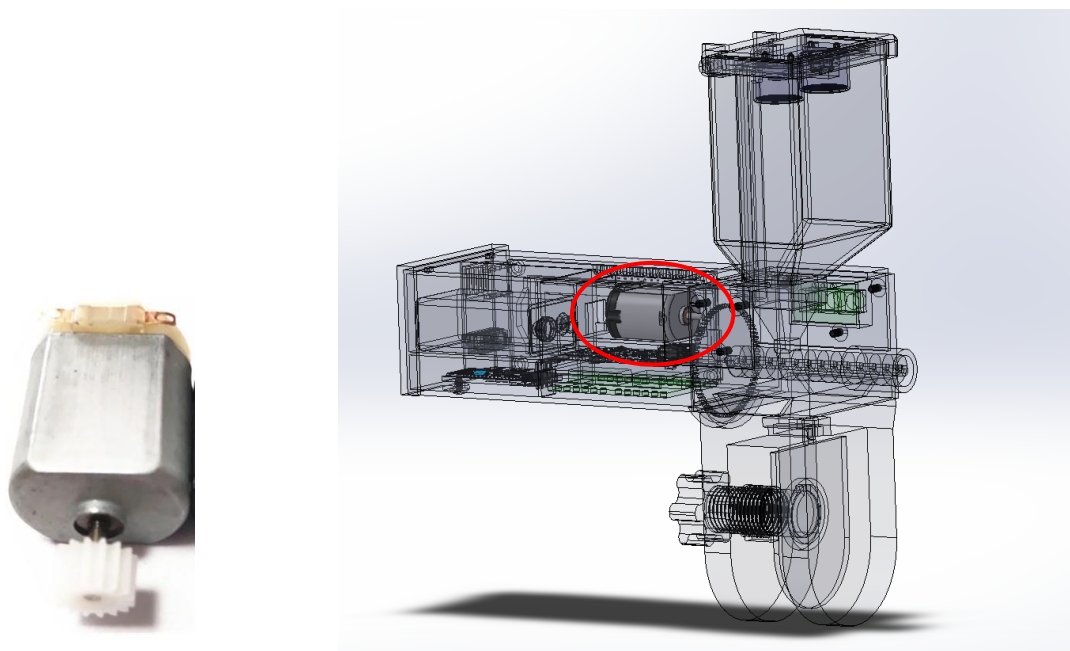
DC motor može raditi s naponom u rasponu od 1.5V do 6V, a pri maksimalnom naponu od 6V postiže 16000 okretaja u minuti (RPM) što ga čini izuzetno brzim za svoju veličinu. Tijekom rada, jakost struje motora ne prelazi 0.3A, stoga je prikladan za upotrebu u malim uređajima gdje nije potrebna velika snaga i koji se napajaju pomoću baterija. Zbog svoje kompaktne veličine i velike brzine, glavni je kandidat za izbor motora za sitnu elektroniku. Može se lako povezati s jednostavnim upravljačkim sklopovima ili mikrokontrolerima (Arduina). Ovaj motor također omogućuje prilagodbu brzine i smjera okretaja ukoliko se preko odgovarajućeg kontrolera poveže s Arduinom pružajući fleksibilnost u postizanju željene brzine i smjera za specifične primjene. Također, karakterizira ga pristupačna cijena od oko 0,5 dolara [18].

Primjena DC motora kod hranilice jest pokretanje vijka za dobavu hrane.

Spajanje DC motora izvršeno je na slijedeći način:

+ pol – OUTPUT1 pin na L293D motor kontroleru

- pol – OUTPUT2 pin na L293D motor kontroleru



Slika 3.8. Mini DC motor i njegov položaj u sklopu [18]

Tablica 3.5. Specifikacije Mini DC motora

Okretni moment	0,6374 N/mm	Stalna struja	0,3A/50W
Konstrukcija	Permanentan magnet	Radni napon	1,5 – 6V
Brzina okretaja	16000 RPM		

3.2.6. 5 – segmentni LED zaslon / kontrolna lampica

5 – segmentni LED zaslon / kontrolna lampica je jednostavan elektronički modul dizajniran kao statusna traka pomoću pet crvenih LED segmenata. Zaslon se sastoji od pet LED segmenata koji se mogu pojedinačno kontrolirati kako bi se prikazale različite vrijednosti ili razine što ga čini idealnim za prikaz jednostavnih podataka poput razine signala. Crvena boja LED dioda je dobro vidljiva i pri slabijem osvjetljenju što doprinosi jasnoći prikaza. Modul je postavljen na elektroničku PCB ploču koja omogućuje lako povezivanje s Arduino platformom. Svaki LED segment se kontrolira putem zasebnog pina što omogućuje preciznu kontrolu nad prikazom. Također, alternativni način spajanja je preko shift registra čime se postiže ušteda na broju pinova s kojim se zaslon povezuje s Arduino. Cijenom od 0,15 dolara po komadu pristupačna je opcija za raznolik spektar primjena [19].

Primjena LED zaslona kod hranilice jest signalizacija korisniku o stanju hrane u spremniku.

Spajanje LED zaslona izvršeno je na slijedeći način:

1 pin ulaz – Q0 pin na 8 – bitnom shift registru

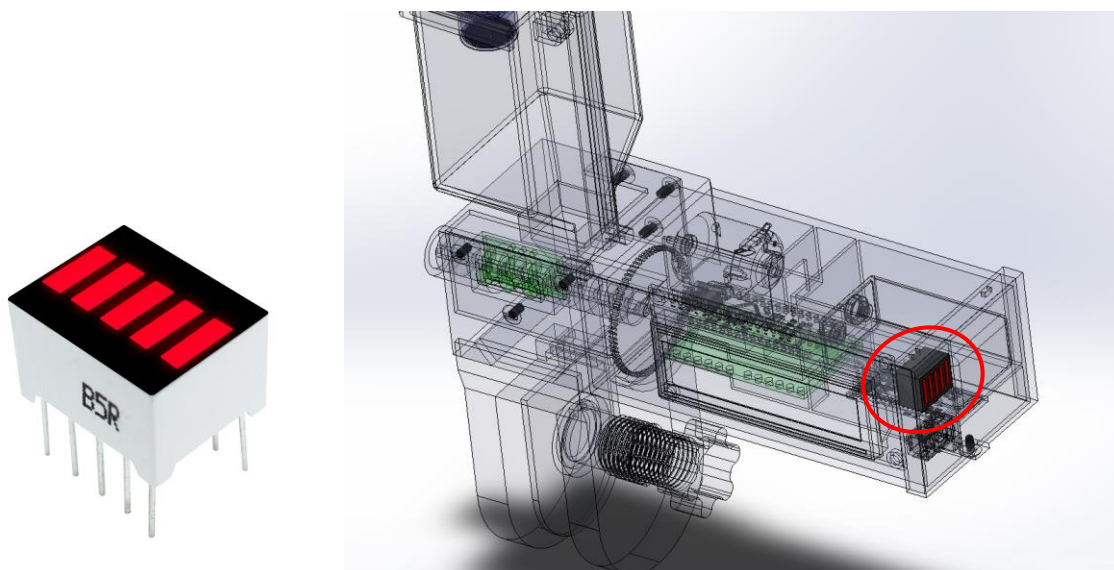
2 pin ulaz – Q1 pin na 8 – bitnom shift registru

3 pin ulaz – Q2 pin na 8 – bitnom shift registru

4 pin ulaz – Q3 pin na 8 – bitnom shift registru

5 pin ulaz – Q4 pin na 8 – bitnom shift registru

svi pinovi izlaz – GND pin na Arduino (-)



Slika 3.9. 5 – segmentni LED zaslon i njegov položaj u sklopu [19]

Tablica 3.6. Specifikacije 5 – segmentnog LED zaslona

Radni napon	oko 2V	Jačina svjetlosti	1000 - 3000 mcd
Radna struja	10-20A		

3.2.7. *Tipkala / tipkovnički modul*

Ovaj modul se sastoji od triju tipkala postavljenih na PCB pločicu koja omogućuje jednostavnu integraciju u elektroničke projekte. Tipkala su raspoređena linearno, a svako tipkalo može biti povezano na različite ulazne pinove mikrokontrolera kako bi se detektirao pritisak na isto. Modul je vrlo jednostavan za povezivanje s mikrokontrolerima te je posebno pogodan zbog svoje kompaktne veličine [20].

Primjena tipkala kod hranilice jest pružanje korisniku mogućnost za namještanje parametara hranilice

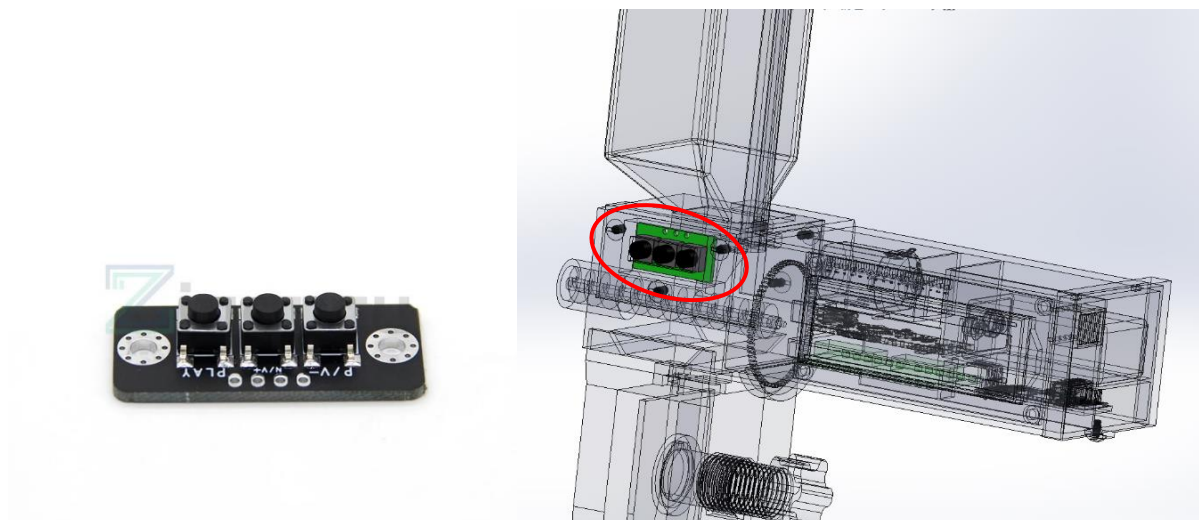
Spajanje tipkala izvršeno je na slijedeći način:

KCOM pin – GND pin na Arduinu (-)

1 pin – pin 2 na digitalnom ulazu Arduina

2 pin – pin 8 na digitalnom ulazu Arduina

3 pin – pin 9 na digitalnom ulazu Arduina



Slika 3.10. Tipkala i njihov položaj u sklopu [20]

Tablica 3.7. Specifikacije tipkala

Radni napon	3.3 – 5V	Max. struja	50 – 100 mA
Tip kontakta	Trenutni kontakt	Vijek trajanja	500000 ciklusa
Otpornost kontakta	100 mΩ	Dimenzije	35 14 mm

3.2.8. Baterija 9V

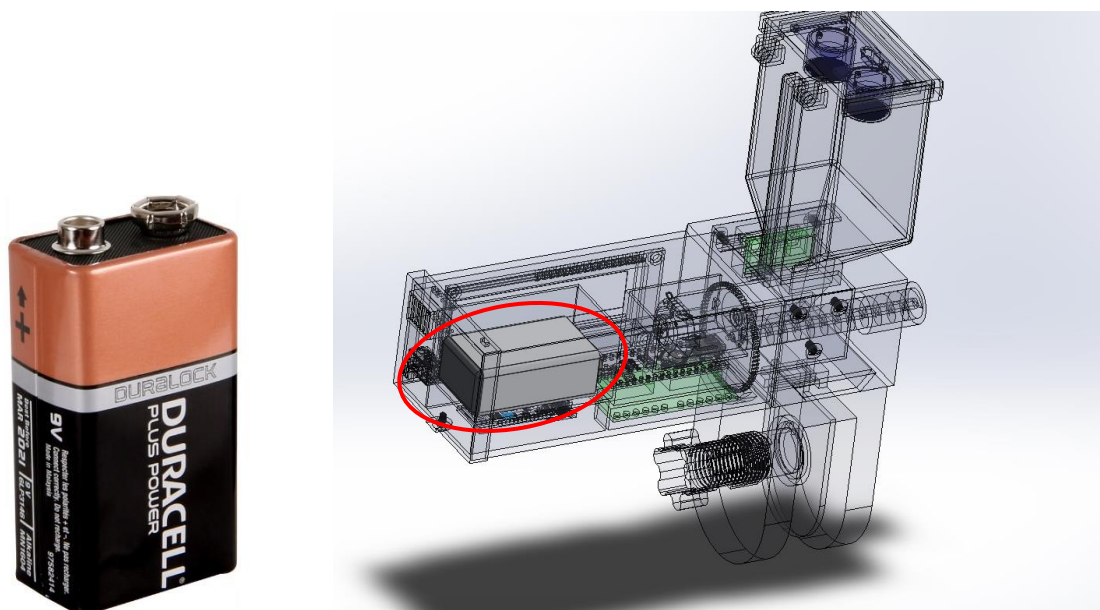
9V litijumska baterija je jedna od najprisutnijih vrsta baterija koja je često korištena u raznim elektroničkim uređajima zbog svog relativno visokog napona, konstantne struje, kompaktnog dizajna i velikog kapaciteta. Baterija sadrži šest serijskih spojenih ćelija od po 1.5 V što daje ukupno 9 V. Kapacitet baterije jest oko 1200 mAh, omogućavajući relativno dugo trajanje (ovisno o namjeni). Često se koristi u kombinaciji s baterijskim konektorom ili kućištem koje olakšava njezino integriranje bez dodatnog alata za montažu ili zamjenu. Baterija može biti i punjiva. Cijena iznosi oko 5 dolara [21].

Primjena baterije kod hranilice jest napajanje elektroničkog sklopa.

Spajanje baterije izvršeno je na slijedeći način:

+ pin – preko sklopke na VIN pin na Arduinu

- pin – GND pin na Arduinu (-)



Slika 3.11. Baterija od 9V i njezin položaj u sklopu [22]

Tablica 3.8. Specifikacije baterije od 9V

Napon	9 V	Dimenzije	48 x 26 x 17 mm
Kapacitet	1200 mAh	Masa	46 grama
Struja	15-50 mA	Vrsta terminala	Snap

3.2.9. Žice i konektor za bateriju

Žice su elementi el. sklopova koje služe za međusobno povezivanje i umrežavanje komponenti tj. svrha im je prijenos signala u obliku struje/napona. Njihova fleksibilnost omogućuje jednostavno povezivanje različitih modula, senzora, i drugih komponenti.

Konektor za 9 V bateriju omogućuje sigurno i jednostavno povezivanje baterije s ostatkom sklopa. Ima dva izlazna vodiča, crveni za pozitivan i crni za negativan pol, koji se mogu lako spojiti na Arduino. Ovakvi konektori su izuzetno praktični jer omogućuju brzo i jednostavno spajanje i odspajanje baterije bez potrebe za lemilicom ili drugim alatima.



Slika 3.12. Konektor za bateriju od 9V [23]

Tablica 3.9. Specifikacije izoliranih žica promjera 1,36mm

Vanjski promjer	1,36 mm	Materijal žice	Bakar
Promjer žice	1 mm – 12*0,1 mm	Temp. otpornost	Do 105 °C

3.2.10. L293D motor kontroler

L293D motor kontroler je integrirani krug (IC) koji se koristi za upravljanje malim motorima. On omogućuje jednostavno upravljanje motorima na način da omogućuje kontroliranje smjera vrtnje i brzine jednog ili dva motora istovremeno. Budući da je kontroler „H-bridge motor driver“, to znači da može mijenjati smjer struje kroz motor omogućujući tako promjenu smjera rotacije. Ovaj kontroler može upravljati motorima s naponom između 4.5 V i 36 V, a svaki kanal može isporučiti struju do 600 mA s maksimalnim vršnim strujnim opterećenjem od 1.2 A po kanalu. To ga čini pogodnim za većinu malih DC motora. Jedna od njegovih ključnih prednosti je njegova unutarnja zaštita od povratne struje što znači da se motori mogu sigurno zaustaviti ili promijeniti smjer bez rizika od oštećenja samog kontrolera ili drugih komponenti u sustavu. L293D je jednostavan za korištenje, posebno u kombinaciji s Arduinom, a i izuzetno je pristupačan svojom cijenom od oko 0,30 dolara [24].

Primjena L293D motor kontrolera kod hranilice jest pružanje mogućnosti kontrole brzine i smjera vrtnje motora.

Spajanje L293D motor kontrolera izvršeno je na slijedeći način:

OUTPUT1 pin – + pol na DC motoru

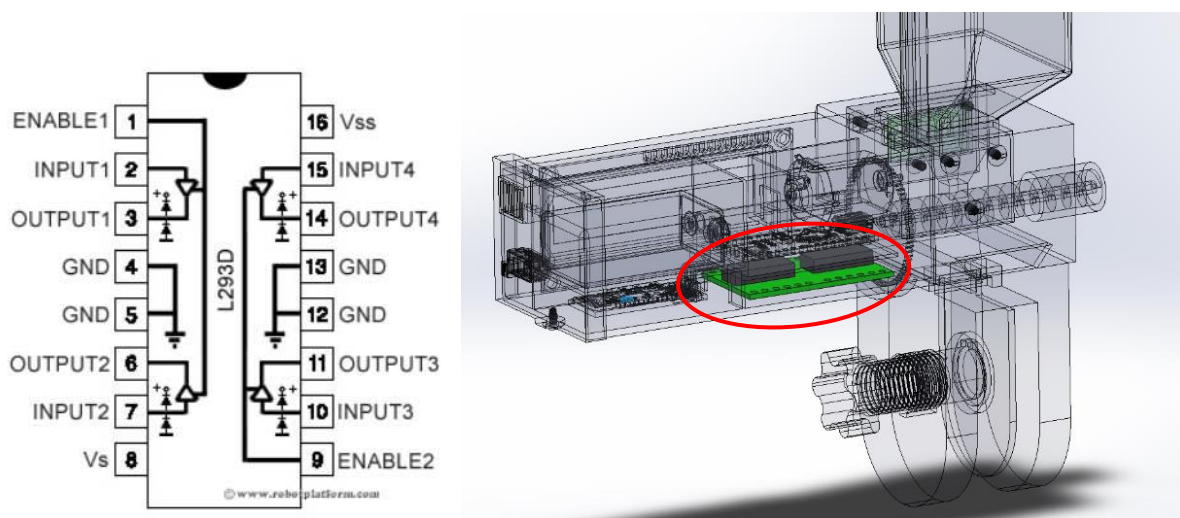
OUTPUT2 pin – - pol na DC motoru

GND pin – GND pin na Arduinu (-)

Vs pin – napajanje od 5V na Arduinu

ENABLE1 pin – pin 3 na digitalnom ulazu Arduina

INPUT1 pin – pin 4 na digitalnom ulazu Arduina



Slika 3.13. L293D motor kontroler i njegov položaj u sklopu [25]

Tablica 3.10. Specifikacije L293D motor kontrolera

Raspon napona	4.5 - 36 V	Izlazna struja	600 mA po kanalu
Napajanje ulazne logike	Odvojeno	Maksimalna izlazna struja	1.2 A po kanalu
Zaštita	Unutarnja ESD zaštita	Vrste dioda	Izlazne stezne diode za induktivno suzbijanje prijelaza
Otpornost na šum	High-Noise-Immunity Inputs		

3.2.11. 8-bitni shift registar

8-bitni shift registar je digitalni sklop koji se koristi za serijsko pomicanje (shiftanje) podataka kroz niz registara te često za proširenje broja izlaznih pinova mikrokontrolera. Shift registar uzima podatke serijski tj. bit po bit i prebacuje ih u paralelni izlaz ili obrnuto. To znači da se, na primjer, može unijeti 8 bitova podataka serijski putem jednog pina, a zatim ih istovremeno pročitati preko 8 paralelnih izlaznih pinova. Jedna od najčešće korištenih vrsta 8-bitnih shift registara je SN74HC595N. On koristi tri glavna pina za rad: podatkovni ulaz (DS), taktni ulaz (SH_CP), i ulaz za zaključavanje (ST_CP). Pomoću ovih pinova podaci se pomiču kroz registar sa svakim taktним impulsom, a nakon što su svi bitovi uneseni, mogu se zaključati i prikazati na paralelnim izlazima. Najčešća primjena je kontrola LED dioda. Ovi registri su izuzetno korisni za pojednostavljenje složenih digitalnih sustava, a i pristupačni su svojom cijenom od 0,10 dolara [26].

Primjena 8-bitnog shift registra kod hranilice jest kontrola 5-segmentnog LED zaslona.

Spajanje 8-bitnog shift registra izvršeno je na slijedeći način:

Q0 pin – 1 pin ulaz na LED zaslonu

Q1 pin – 2 pin ulaz na LED zaslonu

Q2 pin – 3 pin ulaz na LED zaslonu

Q3 pin – 4 pin ulaz na LED zaslonu

Q4 pin – 5 pin ulaz na LED zaslonu

GND pin – GND pin na Arduino (-)

OE pin – GND pin na Arduino (-)

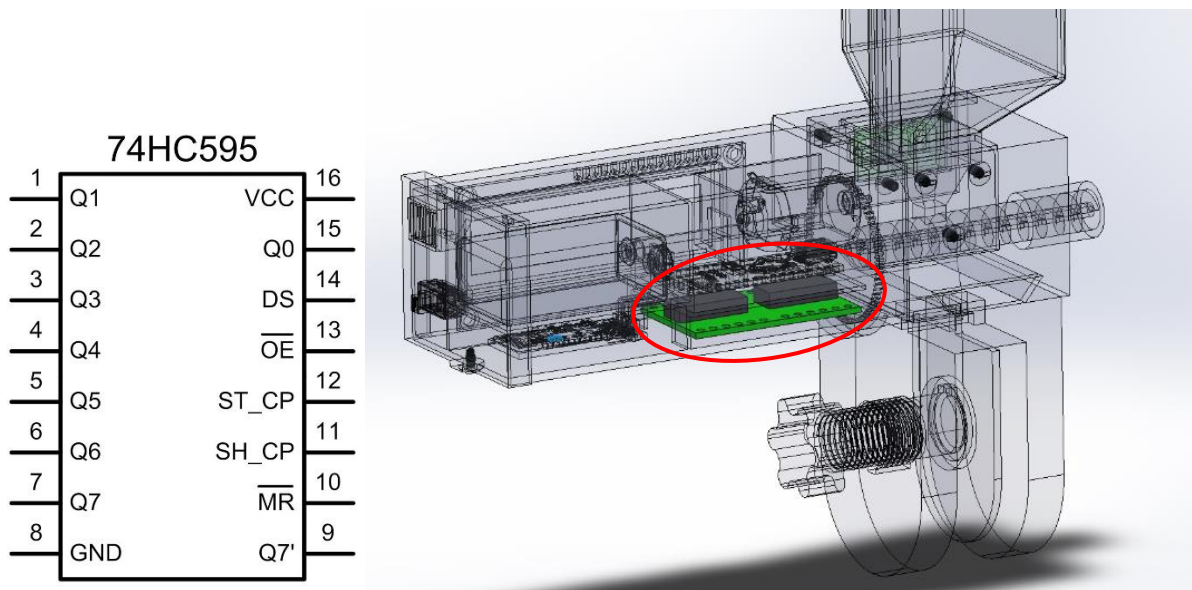
VCC pin – napajanje od 5V na Arduino

MR pin – napajanje od 5V na Arduino

SH_CP pin – pin 13 na digitalnom ulazu Arduina

ST_CP pin – pin 7 na digitalnom ulazu Arduina

DS pin – pin 12 na digitalnom ulazu Arduina



Slika 3.14. 8-bitni shift registar i njegov položaj u sklopu [27]

Tablica 3.11. Specifikacije 8-bitni shift registra

Broj bitova	8-bitni registar	Maksimalna ukupna izlazna struja	70 mA
Napajanje (Vcc)	2V do 6V	Clock Frequency	Do 25 MHz
Ulazna logička razina	Visoka (VIH): Min 2V (pri 5V Vcc). Niska (VIL): Max 0.8V (pri 5V Vcc).	Propagacijsko kašnjenje	Oko 20 ns
Maksimalna izlazna struja po pinu	20 mA	Radna temperatura	-40°C do 125°C

4. PROGRAMIRANJE

Da bi mikrokontroler na Arduino pločici imao sposobnost upravljanja procesom, potrebno je s računala prenijeti kod koji se izvršava na njemu. Isto tako, potrebno je izvršiti programiranje i dizajn aplikacije s kojom se preko Bluetooth modula upravlja procesom. U ovom poglavlju dano je detaljno objašnjenje oba programa. Osnovni preduvjet za uspješno programiranje jest jasno definiranje zadataka koje pojedina komponenta treba izvršiti tj. jasno definirati opis rada sustava.

4.1. Detaljan opis rada

Temelj sustava čini razvojna pločica Arduino Nano na kojoj se nalazi mikrokontroler ATmega328P (iako se simulacija provodila s razvojnom pločicom Arduino Uno radi jednostavnosti i lakšeg testiranja, obje pločice nude skoro identične karakteristike samo u drukčijim dimenzijskim okvirima). Na razvojnu pločicu je putem USB kabela prebačen program s računala koji se procesira u mikrokontroleru i upravlja procesom. Iako bi u stvarnom modelu Arduino Nano bio napajan pomoću baterije (smještene u kućište) od 9V (spojene na GND i VIN pinove preko sklopke smještene u poklopac kućišta), na simulaciji to nije slučaj (radi jednostavnosti i praktičnosti) te je Arduino također napajan putem USB kabela kojim je spojen na računalo. Hrana za ribice pohranjena je u gravitacijskom spremniku koji se nalazi na gornjem dijelu hranilice. Hrana se pomoću gravitacije, kroz dobavni kanal u tijelu hranilice koji je u obliku lijevka, usmjerava u kanal gdje se nalazi puzni vijak za dobavu hrane koji ima spiralni navoj i svojim okretanjem dovodi hranu do izlaza iz kanala tj. do akvarija. Na poklopcu gravitacijskog spremnika smješten je ultrazvučni senzor koji neprekidno pomoću ultrazvučnih valova mjeri udaljenost površine hrane u spremniku od sebe te na temelju toga se zaključuje količina hrane u spremniku. Žice se do kućišta hranilice (a time i do Arduina) dovode kroz kanalice u spremniku za hranu i glavnoj nosivoj konstrukciji. Kako bi se vlasniku dalo do znanja kolika je količina hrane u hranilici, na poklopcu kućišta pozicioniran je 5 – segmentni LED zaslon. Broj segmenata koji svijetle ovise o količine hrane u spremniku i to po sistemu: 5 segmenata: 100 – 80 % napunjenosti, 4 segmenta: 80 – 60 % napunjenosti, 3 segmenta: 60 – 40 % napunjenosti, 2 segmenta: 40 – 20 % napunjenosti, 1 segment: 20 – 0 % napunjenosti. Zaslon je s Arduinom povezan preko 8 - bitnog shift registra koji se nalazi u

kućištu hranilice. Također, LED zaslon svijetli cijelo vrijeme dok je hranilica uključena iz razloga što je količina hrane u spremniku najbitniji podatak za korisnika (nema hrane = životna opasnost po ribe u akvariju). Vijak za dobavu hrane okreće se pomoću DC motora pozicioniranog u kućištu hranilice. Motor je s vijkom povezan preko zupčanika u omjeru 1:6. Taj omjer omogućava veći moment i manju brzinu na vijku, a što su poželjne karakteristike budući da DC motor na svom vratilu daje veliku brzinu i mali moment što otežava kontrolu količine i okretanje vijka. Motor je s Arduinoom povezan preko L293D motornog kontrolera koji omogućava upravljanje brzinom motora i smjerom njegove vrtnje. Motorni kontroler i shift registar se na realnom modelu nalaze na istoj tiskanoj pločici pozicioniranoj ispod Arduina u kućištu hranilice dok se na simulaciji nalaze na razvojnoj pločici s ostatkom komponenata. Kako bi korisnik hranilice mogao namještati parametre hranilice i time kontrolirati proces, na kućištu hranilice je pozicioniran 16x2 LCD zaslon upravljan pomoću 3 tipkala koja se nalaze u udubini glavne nosive konstrukcije hranilice. Postoje dva parametra koji se podešavaju, a to su vremenski razmak između usipavanja hrane (u danjem tekstu: vrijeme) i količina hrane koja se usipava u akvarij svaki put kada se motor pokrene tj. dužina intervala rada motora (u danjem tekstu: količina). Odabrani vremenski intervali su: 4, 8, 12, 16, 20 i 24 sata dok su odabrane količine hrane 0.4, 0.6, 1, 1.8, 2 i 3.6 grama. Takvim spektrom vremena i količina pokrivena su potrebe za praktički sve uobičajene akvarijske ribice. Tipkala kojima se upravlja proces jesu MODE tipkalo koje se mijenja prikaz na ekranu između vremena i količine, + tipkalo koje prebacuje varijablu na prvu veću vrijednost (ovisno o MODE-u u kojem se nalazi) i – tipkalo koje prebacuje varijablu na prvu manju vrijednost (ovisno o MODE-u u kojem se nalazi). LCD zaslon je preko I2C sučelja povezan s Arduinoom dok su tipkala izravno povezana žicama koje do kućišta dolaze preko kanalice u glavnoj nosivoj konstrukciji hranilice. Ako se detektira da korisnik nije pritisnuo niti jedno tipkalo 15 sekundi, gasi se pozadinsko svjetlo na LCD zaslonu iz razloga što nije nužno da korisnik konstantno ima informacije s zaslona na uvidu, a time se štedi baterija. Kako bi se korisniku pružila mogućnost da hranilicom upravlja pomoću aplikacije na svom mobilnom telefonu, u kućište hranilice je također smješten Bluetooth modul izravno povezan s Arduinoom. Pomoću aplikacije korisnik može namještati parametre hranilice te su informacije o količini i vremenu azurne s onima na LCD zaslonu na kućištu hranilice. Također, na aplikaciji korisnik može dobiti detaljniji uvid u stanje hrane u spremniku koji je izražen u postocima, a boja prikaza se mijenja u ovisnosti o količini hrane u spremniku: od 100% od 66% zelena boja, od 66% od 33% žuta boja i od 33% do 0% crvena boja.

4.2. Arduino kod

Kod za Arduino pisan je u programu Arduino IDE (engl. Arduino Integrated Development Enviroment). Taj program je razvojno okruženje specifično dizajnirano za rad s Arduino mikrokontrolerima. On omogućava pisanje, testiranje i učitavanje koda na Arduino razvojne pločice, koristeći programski jezik temeljen na C/C++ programskom jeziku. Budući da program dolazi s velikim brojem ugrađenih biblioteka, olakšava se rad s različitim senzorima, aktuatorima i drugim komponentama. Aktualna verzija programa je 2.3.2.

4.2.1. Deklaracija varijabli 1. dio – definiranje pinova i biblioteke

U ovom dijelu koda pozivaju se Arduino biblioteke koje se koriste u programu za kontrolu pojedinih komponenti (konkretno Bluetooth modula i I2C sučelja kod LCD zaslona). Također, definiraju se pinovi na koje su spojene komponente.

```
1 #include <SoftwareSerial.h> //Uključivanje biblioteke za komunikaciju s bluetooth modulom preko serijskih digitalnih portova
2 #include <Wire.h> // Uključivanje biblioteke za I2C komunikaciju
3 #include <LiquidCrystal_I2C.h> // Uključivanje biblioteke za I2C LCD
4
5
6 LiquidCrystal_I2C lcd(0x27, 16, 2); //postavljanje adrese LCD-a na 0x27 te određivanje dimenzija ekrana
7
8 //PINOVI
9 // Bluetooth pinovi
10 int rxPin = 11; //rx bluetooth pin
11 int txPin = 10; //tx bluetooth pin
12 SoftwareSerial bluetooth(rxPin, txPin);
13
14 // Pinovi za fizičke tipke
15 int modetipkalo = 2; // pin za tipkalo za promjenu prikaza
16 int plustipkalo = 8; // pin za + tipkalo
17 int minustipkalo = 9; // pin za - tipkalo
18
19 // Pinovi za motor
20 int speedpin = 3; // pin za određivanje brzine motora
21 int dir1 = 4; //pin za smijer motora
22
23 // Pinovi za UZ senzor
24 int TRIG_PIN = 6; //pin za odasiljanje signala
25 int ECHO_PIN = 5; // pin za primanje signala
26
27 // Pinovi za signalnu lampicu
28 int latchpin = 7;
29 int clockpin = 13;
30 int datapin = 12;
```

Slika 4.1. Deklaracija varijabli 1. dio – definiranje pinova i biblioteke

4.2.2. Deklaracija varijabli 2. dio – definiranje vrijednosti

Postavljane konstantnih vrijednosti koje olakšavaju snalaženje u kodu te olakšavaju eventualno naknadno mijenjanje vrijednosti. Također, deklariranje početnih vrijednosti za `millis()` funkcije koje se kasnije mijenjaju kroz program, te definiranje nizova koji predstavljaju vrijeme i količinu.

```
34 //POČETNO POSTAVLJENE VRIJEDNOSTI
35 // Vrijednosti za motor
36 int brzina = 255; //brzina motora, između 0 i 255
37
38 // Vrijednosti za ekran
39 int mode = 0; // vrijednost za praćenje stanja ekrana, dali je postavljen na vrijeme ili na količinu
40 int IndexKolicine = 0; // količina
41 int IndexVremena = 0; // vrijeme
42
43 // Vrijednosti za millis za vrijeme vrtnje motora
44 unsigned long VrijemeIskljucenogMotora = 0; // vrijeme za koje motor ne radi - dolazi iz postavljenog vremena
45 unsigned long VrijemeUkljucenogMotora = 0; // vrijeme za koje motor radi - dolazi iz postavljene količine
46 unsigned long PredhodniMillis = 0; // početna vrijednost prethodnog vremena za vrtnju motora
47
48 // Nizovi s mogućim vrijednostima za količinu i vrijeme
49 float kolicina[] = {0.4, 0.6, 1, 1.8, 2, 3.6}; // niz za moguće vrijednosti količine
50 int vremena[] = {4, 8, 12, 16, 20, 24}; // niz za moguće vrijednosti vremena
51
52 // Vrijednosti za signalne lampice
53 byte led = 0xFF; //postavljamo brojeve ledice kao hex, sve su ledice ON
54
55 // Vrijednosti za millis za vrijeme gašenje LCD-a
56 unsigned long PoslijednjiPritisakMillis = 0; // Vžvrijeme zadnjeg pritiska tipkala
57 unsigned long LCDVrijeme = 15000; // vrijeme koje lcd svijetli
58
59 // Vrijednost visine hranilice
60 float DimHranilice = 19; // visina hranilice u cm
```

Slika 4.2. Deklaracija varijabli 2. dio – definiranje vrijednosti

4.2.3. Inicijalizacijska rutina (void setup)

Definiranje serijske komunikacije (sa serijskim (engl. serial) monitorom na računalu) te komunikacije s Bluetooth modulom. Postavljanje pinova kao ulazni ili izlazni te definiranje tipkala kao pullup (detektira se uzlazni brid prilikom pritiska). Uključivanje LCD zaslona i njegova početna inicijalizacija.

```
62 void setup() {
63     Serial.begin(9600); // Postavljanje komunikacije s serial monitorom
64
65     bluetooth.begin(9600); // Postavljanje komunikacije s bluetooth modulom
66
67     pinMode(modetipkalo, INPUT_PULLUP); // postavljanje tipkala za promjenu prikaza kao pullup tipkalo
68     pinMode(plustipkalo, INPUT_PULLUP); // postavljanje + tipkala kao pullup tipkalo
69     pinMode(minustipkalo, INPUT_PULLUP); // postavljanje - tipkala kao pullup tipkalo
70
71     pinMode(speedpin, OUTPUT); // postavljanje pina za brzinu kao izlaznog
72     pinMode(dir1, OUTPUT); // postavljanje pina za smjer kao izlaznog
73
74     pinMode(latchpin, OUTPUT); //postavljanje svih parametara signalne lampice kao izlazne
75     pinMode(datapin, OUTPUT);
76     pinMode(clockpin, OUTPUT);
77
78     pinMode(TRIG_PIN, OUTPUT); // postavljanje pina za odašiljanje kao izlaznog
79     pinMode(ECHO_PIN, INPUT); // Postavljanje pina za primanje kao ulaznog
80
81
82     lcd.init(); // postavljanje LCD-a
83     lcd.backlight(); // uključivanje pozadinskog osvjetljenja LCD-a
84     displayAmount(); // prikaz količine na LCD-u kao prvog podatka pri uključivanju
85 }
```

Slika 4.3. Inicijalizacijska rutina (void setup)

4.2.4. Petlja (void loop) 1. dio – Kontrola količine hrane

Ovdje počinje dio koda koji se konstantno vrti u krug. U prvom dijelu koda izvršava se dio kontrole količine hrane u spremniku pomoću ultrazvučnog senzora te signaliziranje stanja iste pomoću 5 – segmentnog LED zaslona.

```

88 void loop() {
89     //KONTORLA KOLIČINE HRANE U HRANILICI
90     // Postavljanje čipa za kontrolu signalne lampice
91     digitalWrite(latchpin, LOW); // podaci dolaze
92     shiftOut(datapin, clockpin, LSBFIRST, led); // bitan je točno taj redosljed
93     digitalWrite(latchpin, HIGH); // podaci odlaze
94
95     // Postavljanje pina za odašiljanje UZ senzora na nisko stanje na 2 ms da bismo očistili senzor smetnji
96     digitalWrite(TRIG_PIN, LOW);
97     delayMicroseconds(2);
98
99     // Slanje UZ impulsa
100    digitalWrite(TRIG_PIN, HIGH); // slanje impulsa u trajanju 10 ms
101    delayMicroseconds(10);
102    digitalWrite(TRIG_PIN, LOW); // prekidanje slanja impulsa
103
104
105
106    long trajanje = pulseIn(ECHO_PIN, HIGH); // mjerenje trajanja signala na pinu za primanje
107
108    float udaljenost = trajanje / 2.0 / 29.1; // udaljenost hrane od senzora u cm
109
110    float postotak = (udaljenost / DimHranilice) * 100; // postotak napunjenosti hranilice
111
112    // Prikazivanje postotka napunjenosti pomoću signalne lampice
113    if (postotak > 80){
114        | led = 0xF8; // uključeno je svih 5 ledica
115    }
116    if (postotak > 60 && postotak < 80){
117        | led = 0xF0; // uključene su 4 ledice
118    }
119    if (postotak > 40 && postotak < 60){
120        | led = 0xE0; // uključene su 3 ledice
121    }
122    if (postotak > 20 && postotak < 40){
123        | led = 0xC0; // uključene su dvije ledice
124    }
125    if (postotak > 2 && postotak < 20){
126        | led = 0x80; // uključena je jedna ledica
127    }

```

Slika 4.4. Petlja (void loop) 1. dio – Kontrola količine hrane

4.2.5. Petlja (void loop) 2. dio – Upravljanje pomoću tipkala

U ovom dijelu petlje definiraju se funkcije triju fizičkih tipkala: MODE, + i -.

```

130 //UPRAVLJANJE PROCESOM POMOĆU FIZIČKIH TIPKALA
131 //Tipkalo za promjenu prikaza
132 if (digitalRead(modetipkalo) == LOW) { // provjera stanja tipkala za promjenu prikaza
133     PoslijednjiPritisakMillis = millis(); // postavljanje novog vremena zadnjeg pritiska tipkala radi gašenja lcd-a
134     lcd.backlight(); // upali lcd
135     delay(200);
136     mode = (mode + 1) % 2; // postavljanje jednadzbe za promjenu stanja prikaza (ima 2 stanja)
137     if (mode == 0) {
138         displayAmount(); // prikaz količine
139     } else {
140         displayTime(); // prikaz vremena
141     }
142     while (digitalRead(modetipkalo) == LOW); // čekanje da se tipkalo otpusti
143 }
144
145 // + tipkalo
146 if (digitalRead(plustipkalo) == LOW) { // provjera stanja + tipkala
147     PoslijednjiPritisakMillis = millis(); // postavljanje novog vremena zadnjeg pritiska tipkala radi gašenja lcd-a
148     lcd.backlight(); // upali lcd
149     delay(200);
150     if (mode == 0) { //ako smo na količinskom prikazu
151         IndexKolicine = (IndexKolicine + 1) % 6; // postavljanje jednadzbe za povećanje stanja (ima 6 stanja)
152         displayAmount(); // prikaz količine
153     } else { // ako smo na vremenskom prikazu
154         IndexVremena = (IndexVremena + 1) % 6; // postavljanje jednadzbe za povećanje stanja (ima 6 stanja)
155         displayTime(); // prikaz vremena
156     }
157     while (digitalRead(plustipkalo) == LOW); // čekanje da se tipkalo otpusti
158 }
159
160 // - tipkalo
161 if (digitalRead(minustipkalo) == LOW) { // provjera stanja - tipkala
162     PoslijednjiPritisakMillis = millis(); // postavljanje novog vremena zadnjeg pritiska tipkala radi gašenja lcd-a
163     lcd.backlight(); // upali lcd
164     delay(200);
165     if (mode == 0) { //ako smo na količinskom prikazu
166         IndexKolicine = (IndexKolicine - 1 + 6) % 6; // postavljanje jednadzbe za smanjenje stanja (ima 6 stanja)
167         displayAmount(); // prikaz količine
168     } else { // ako smo na vremenskom prikazu
169         IndexVremena = (IndexVremena - 1 + 6) % 6; // postavljanje jednadzbe za smanjenje stanja (ima 6 stanja)
170         displayTime(); // prikaz vremena
171     }
172     while (digitalRead(minustipkalo) == LOW); // čekanje da se tipkalo otpusti
173 }

```

Slika 4.5. Petlja (void loop) 2. dio – Upravljanje pomoću tipkala

4.2.6. Petlja (void loop) 3. dio – Upravljanje pomoću aplikacije preko Bluetooth modula

U ovom dijelu petlje definira se upravljanje procesom pomoću signala koje Arduino prima od aplikacije na mobilnom telefonu preko Bluetooth modula.

```

177 //UPRAVLJANJE PROCESOM POMOĆU BLUETOOTH MODULA I APLIKACIJE
178 //Provjera dostupnosti podataka
179 if (bluetooth.available()) {
180   char data = bluetooth.read(); // primanje znaka s aplikacije
181
182   // Digitalna tipkala u aplikaciji s kojima se upravlja proces (komentirati cu samo ono što je različito od fizičkih tipkala)
183   if (data == 'V') { // ako je primljen znak V (char v) - tipkalo se ponaša kao segment fizičkog tipkala za promjenu prikaza
184     PoslijednjiPritisakMillis = millis();
185     lcd.backlight();
186     mode = 1; // postavljanje na vremenski prikaz
187     displayTime();
188   } else if (data == 'K') { // ako je primljen znak K (char K) - tipkalo se ponaša kao segment fizičkog tipkala za promjenu prikaza
189     PoslijednjiPritisakMillis = millis();
190     lcd.backlight();
191     mode = 0; // postavljanje na količinski prikaz
192     displayAmount();
193   } else if (data == '+') { // ako je primljen znak + (char +) - tipkalo se ponaša kao fizičko + tipkalo
194     PoslijednjiPritisakMillis = millis();
195     lcd.backlight();
196     if (mode == 0) {
197       IndexKolicine = (IndexKolicine + 1) % 6;
198       displayAmount();
199     } else {
200       IndexVremena = (IndexVremena + 1) % 6;
201       displayTime();
202     }
203   } else if (data == '-') { // ako je primljen znak - (char -) - tipkalo se ponaša kao fizičko - tipkalo
204     PoslijednjiPritisakMillis = millis();
205     lcd.backlight();
206     if (mode == 0) {
207       IndexKolicine = (IndexKolicine - 1 + 6) % 6;
208       displayAmount();
209     } else {
210       IndexVremena = (IndexVremena - 1 + 6) % 6;
211       displayTime();
212     }
213   }
214 }

```

Slika 4.6. Petlja (void loop) 3. dio – Upravljanje pomoću aplikacije preko Bluetooth modula

4.2.7. Petlja (void loop) 4. dio – Kontrola pozadinskog osvjetljenja LCD zaslona

Kontrola pozadinskog osvjetljenja LCD zaslona pomoću millis() funkcije. Ako nije pritisnuto niti jedan tipkalo iz poglavlja 4.2.5. i 4.2.6. 15 sekundi, pozadinsko svjetlo na zaslonu se gasi radi uštede energije.

```

216 // GAŠENJE SVIJETLA AKO NIJE PRITISNUT TIPKALO
217 if (millis() - PoslijednjiPritisakMillis >= LCDVrijeme) {
218     lcd.noBacklight(); // ugasi lcd ako je prošlo određeno vrijeme mirovanja
219 }

```

Slika 4.7. Petlja (void loop) 4. dio – Kontrola pozadinskog osvjetljenja LCD zaslona

4.2.8. Petlja (void loop) 5. dio – Kontrola rada DC motora

Kontrola motora pomoću millis() funkcije i parametara određenih tipkala u poglavljima 4.2.5. i 4.2.6.

```

221 //MOTOR
222 VrijemeUkljucenogMotora = kolicina[IndexKolicine] * 3600000; // vrijeme u satima
223 VrijemeIskljucenogMotora = vremena[IndexVremena] * 1000; // vrijeme u sekundama (radi testiranja)
224
225
226 unsigned long TrenutniMillis = millis();
227
228 if (TrenutniMillis - PredhodniMillis >= VrijemeIskljucenogMotora) { // Vrijeme za pokretanje motora
229     digitalWrite(dir1, HIGH); // Postavi smjer motora
230     analogWrite(speedpin, mbrzina); // Uključi motor sa zadanom brzinom
231
232     // Provjeri da li je motor dovoljno dugo radio
233     if (TrenutniMillis - PredhodniMillis >= VrijemeIskljucenogMotora + VrijemeUkljucenogMotora) {
234         analogWrite(speedpin, 0); // Isključi motor
235         digitalWrite(dir1, LOW); // Isključi smjer
236
237         PredhodniMillis = TrenutniMillis; // Ponovno postavi PredhodniMillis za sljedeći ciklus
238     }
239 }

```

Slika 4.8. Petlja (void loop) 5. dio – Kontrola rada DC motora

4.2.9. *Petlja (void loop) 6. dio – Slanje podataka na aplikaciju preko Bluetooth modula*

U ovom dijelu koda se šalju informacije o vremenu, količini i stanju količine hrane u spremniku na aplikaciju na mobilnom telefonu preko Bluetooth modula.

```

242 // SLANJE PODATAKA PREKO BLUETOOTH-A
243 bluetooth.print(vremena[IndexVremena]); // ispiši vrijeme na bluetooth ekranu
244 bluetooth.print("|"); // crtica radi odvajanja podataka u programu za aplikaciju
245 bluetooth.print(kolicina[IndexKolicine]); // ispiši količinu na bluetooth ekranu
246 bluetooth.print("|"); // crtica radi odvajanja podataka u programu za aplikaciju
247 bluetooth.println(postotak); // ispiši stanje spremnika za hranu
248 delay(200); //delay - treba biti jer inace ne radi
249 }

```

Slika 4.9. Petlja (void loop) 6. dio – Slanje podataka na aplikaciju preko Bluetooth modula

4.2.10. *Ispisivanje podataka na LCD zaslonu*

Kontrola LCD zaslona i ispisivanje podataka za kontrolu procesom na istome. Kraj Arduino koda.

```

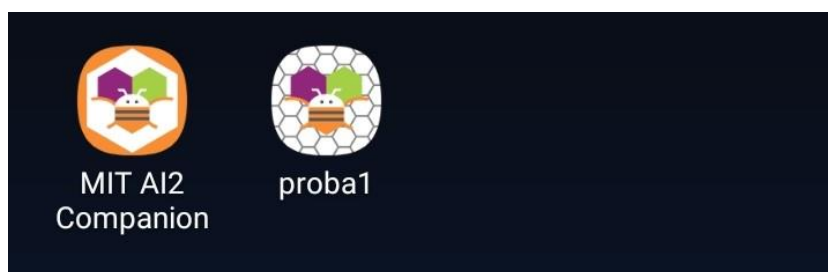
253 // ISPISIVANJE PODATAKA NA LCD DISPLAY-U
254 void displayAmount() {
255     lcd.clear(); // čišćenje LCD ekrana
256     lcd.setCursor(0, 0); // mjesto ispisa na ekranu
257     lcd.print("Kolicina: "); // ispiši količina na LCD-u
258     lcd.setCursor(0, 1); // mjesto ispisa na ekranu
259     lcd.print(kolicina[IndexKolicine]); // ispiši odabranu količinu na LCD-u
260     lcd.print(" grama"); // ispiši grama na LCD-u
261 }
262
263 void displayTime() {
264     lcd.clear(); // čišćenje LCD ekrana
265     lcd.setCursor(0, 0); // mjesto ispisa na ekranu
266     lcd.print("Vrijeme: "); // ispiši vrijeme na LCD-u
267     lcd.setCursor(0, 1); // mjesto ispisa na ekranu
268     lcd.print(vremena[IndexVremena]); // ispiši odabrano vrijeme na LCD-u
269     lcd.print(" sati"); // ispiši sati na LCD-u
270 }

```

Slika 4.10. Ispisivanje podataka na LCD zaslonu

4.3. MIT App Inventor kod

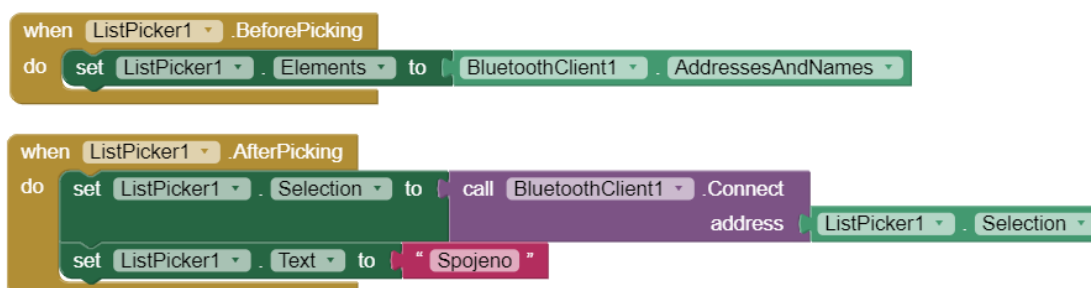
MIT App Inventor je vizualno razvojno okruženje koje omogućava kreiranje aplikacija za Android i iOS uređaje. Razvijen na MIT-u (Massachusetts Institute of Technology), ovaj alat koristi programiranje temeljeno na blokovima koda koji se povezuju u različite funkcionalne blokove kako bi se dizajnirala aplikacija. Sam program se sastoji od dva glavna dijela: dizajnerskog dijela gdje se postavljaju elementi aplikacije i korisničko sučelje, i blokova za programiranje tj. dijela gdje se aplikacija programira. Programu se pristupa preko web stranice (ne preuzima se na računalo), a aplikacija se na mobilni uređaj prebacuje pomoću skeniranja QR koda u aplikaciji MIT AI2 Companion.



Slika 4.11. Izgled ikone aplikacija na početnom zaslonu mobilnog telefona

4.3.1. Povezivanje s Arduinom preko Bluetooth modula

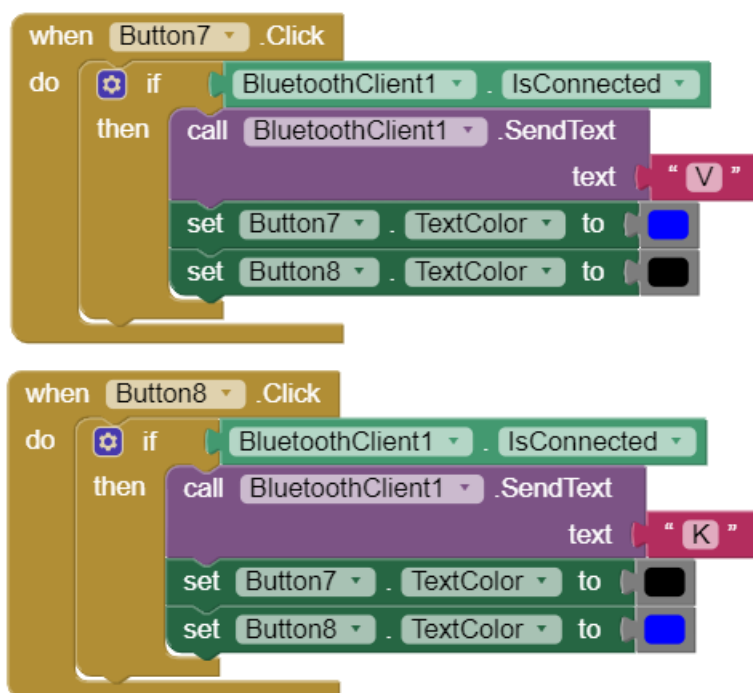
U ovom dijelu blok dijagrama se provjerava dostupnost i vrši odabir bluetooth mreže. Nakon odabira mreže, tekst na tipkalu za odabir se mijenja s „Bluetooth“ na „Spojeno“



Slika 4.12. Povezivanje s Arduinom preko Bluetooth modula

4.3.2. Definiranje tipkala za vrijeme i količinu

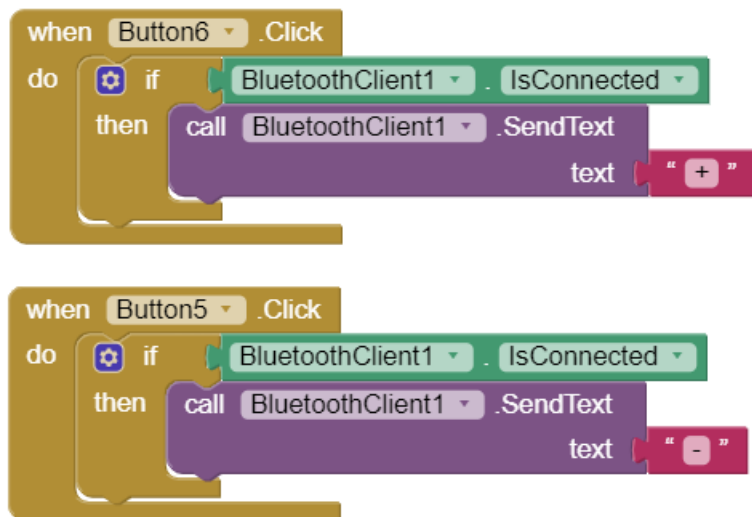
Ako je mobilna aplikacija uspješno spojena na Arduino preko Bluetooth modula omogućuje se kontrola procesom tipkalima za vrijeme i količinu i to na način da tipkalo za vrijeme, kada je pritisnuto, šalje znak „V“ koji se potom u Arduino kodu tumači kao pritisnuto tipkalo, a tipkalo za količinu šalje znak „K“, koji se također tumači u Arduino kodu. Pritisak bilo kojeg tipkala mijenja boju onog nepritisnutog i tako signalizira korisniku u kojem je MODE-u rada.



Slika 4.13. Definiranje tipkala za vrijeme i količinu

4.3.3. Definiranje + i – tipkala

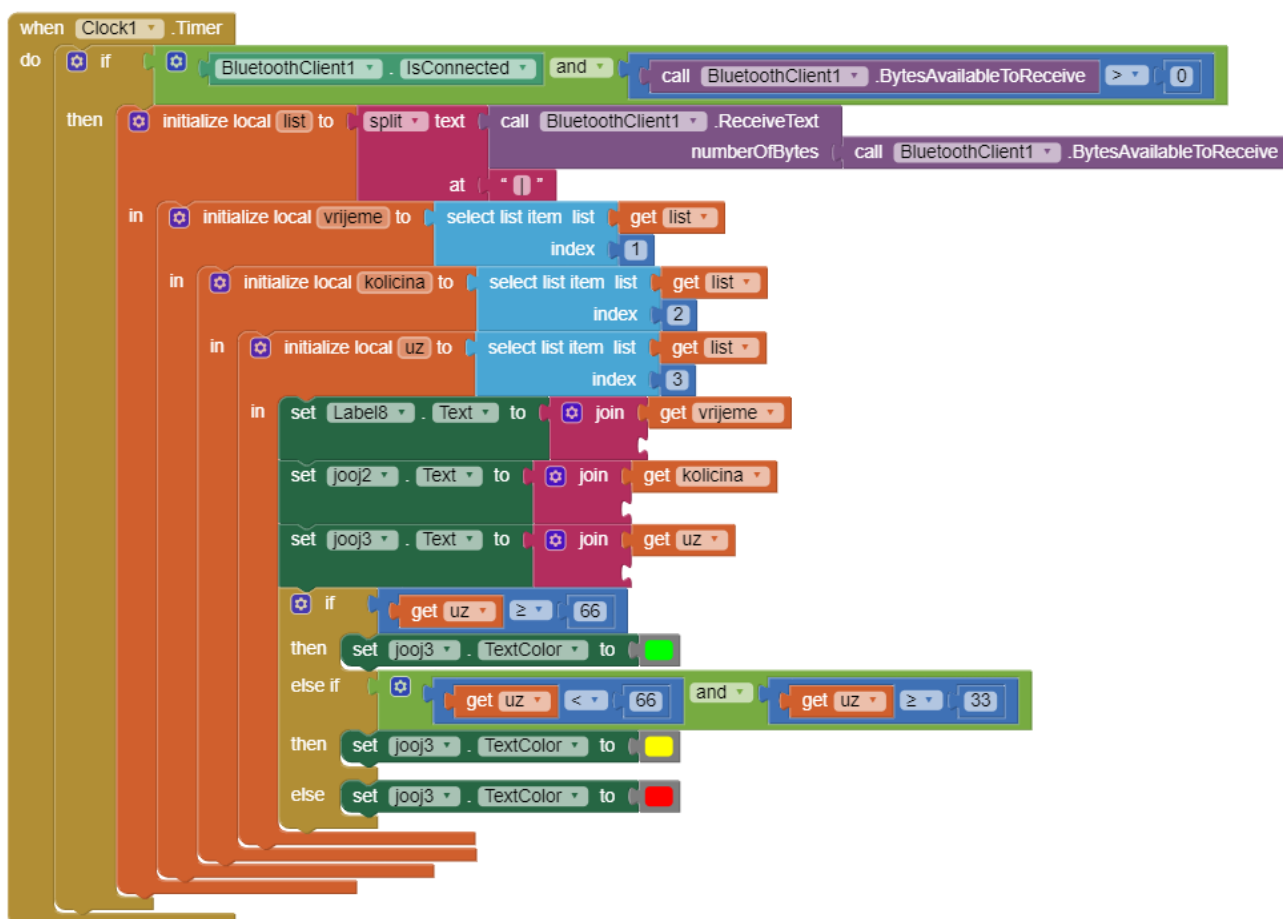
Slično kao i u poglavlju 4.3.2., ako je zadovoljen uvjet povezanosti, šalju se signali u obliku znakova „+“ i „-“ u ovisnosti o pritisnutom tipkalu u aplikaciji.



Slika 4.14. Definiranje + i – tipkala

4.3.4. Primanje podataka s Arduina o vremenu, količini i stanju količine hrane

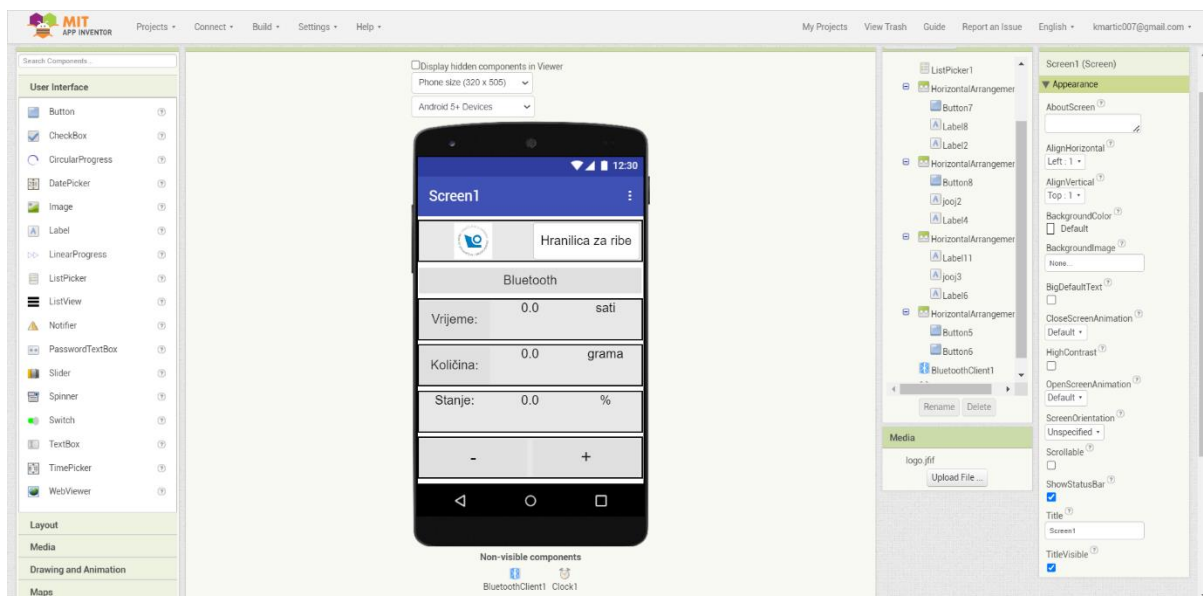
U ovom dijelu koda vrši se obrada podataka koji se šalju s Arduina (preko Bluetooth modula) na mobilni uređaj. Ako je mobilna aplikacija uspješno spojena na Arduino preko Bluetooth modula kreće se s primanjem podataka. Podaci iz poglavlja 4.2.9. šalju se u obliku jedne rečenice unutar koje su podaci odvojeni znakom „|“. Kada se u rečenici detektira taj znak, u kodu dolazi do odvajanja podataka koji se potom ispisuju kao informacije na zasebne zaslone. U zadnjem dijelu programa programira se mijenjanje boje teksta u ovisnosti o količini hrane u spremniku. Cijeli blok se nalazi u virtualnom Clock timer-u koji prima podatke svakih 100ms. Kraj koda.



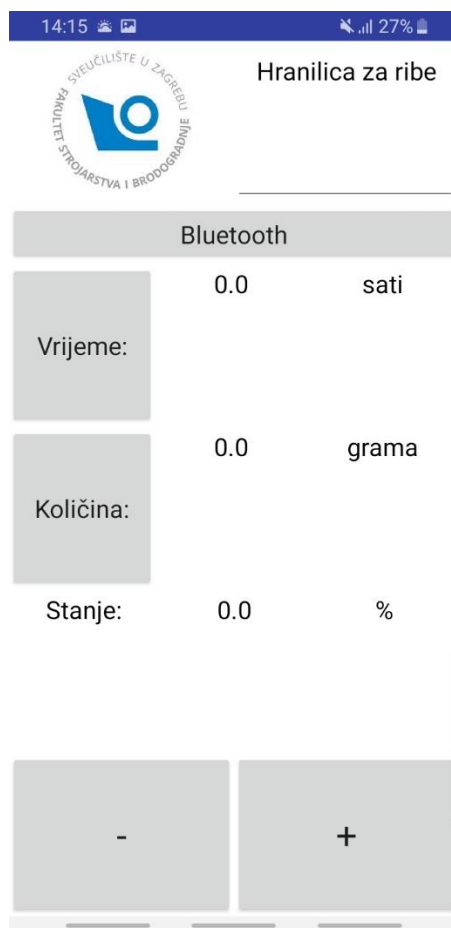
Slika 4.15. Primanje podataka s Arduina o vremenu, količini i stanju količine hrane

4.3.5. Dizajn aplikacije

Početni zaslon aplikacije dizajnira se pomoću povlačenja blokova na zaslon uređaja s odabranim operativnim sustavom (Android ili iOS). Nakon što se komponente smjeste na odabrano mjesto na zaslonu, vrši se prilagođavanje njihovih proporcija na ekranu te određivanje svrhe davanjem imena. Postoje dvije vrste komponenta: vidljive (poput teksta, tipkala itd.) i nevidljivog dijela (poput Bluetootha i Clocka). Razvojno sučelje i gotova aplikacija na mobilnom telefonu vidljivi su na dolje prikazanim slikama.



Slika 4.16. Izgled dizajna aplikacije u programu MIT App Inventor



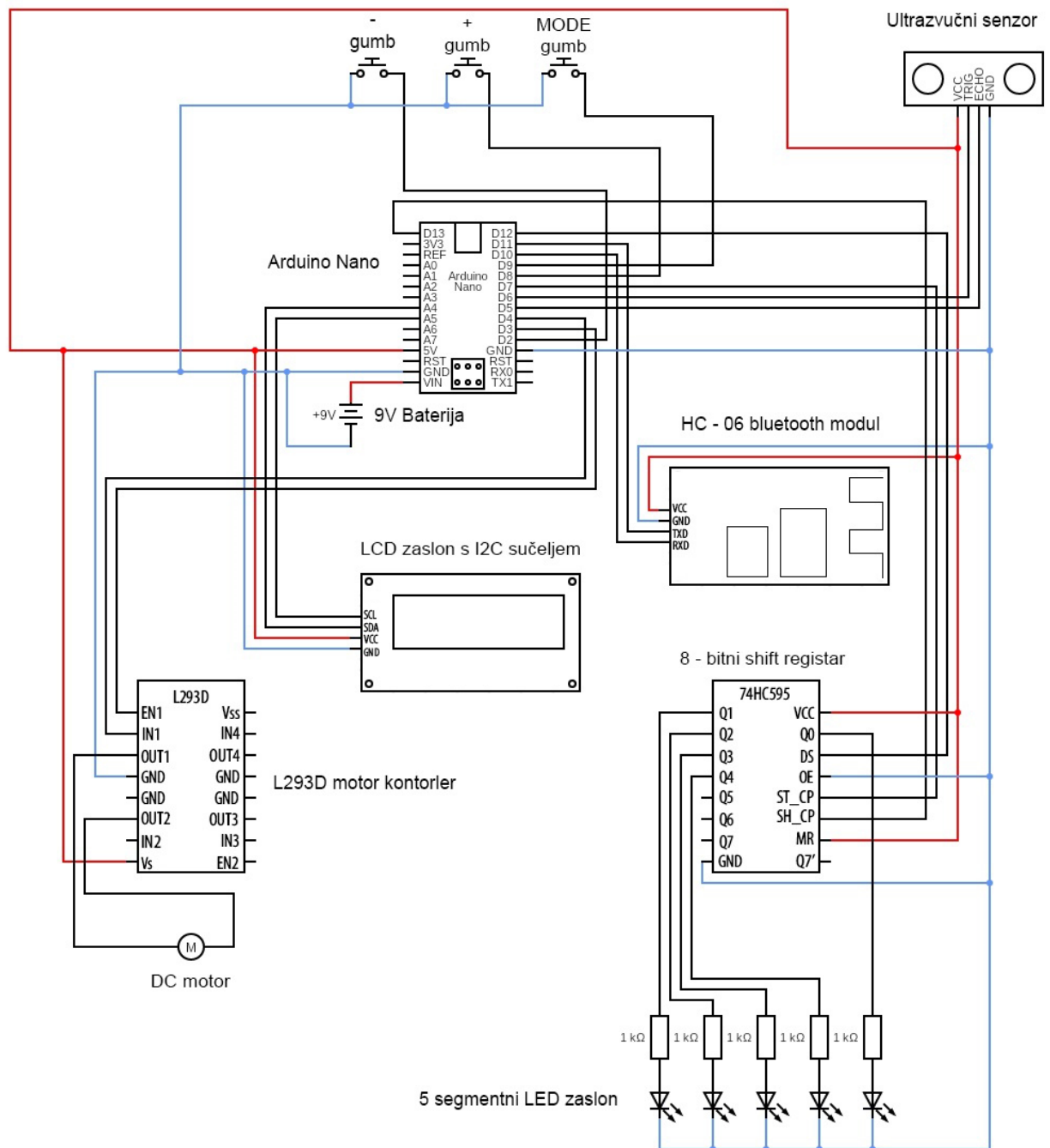
Slika 4.17. Izgled gotove aplikacije koja nije povezana s Arduinoom

5. SPAJANJE I TESTIRANJE (SIMULACIJA)

Nakon što se izvršio odabir komponenti pristupa se spajanju i testiranju rada elektroničkih komponenti u međusobno spojenom odnosu. Testiranje na realnom modelu je važno nakon teorijske provjere i simulacije na računalima iz razloga što uživo postoje faktori koji se na računalnoj simulaciji ne mogu predvidjeti, poput gubitaka u vodovima koji rezultira premalim naponom na komponenti, neispravna komponenta itd.

5.1. Električna shema

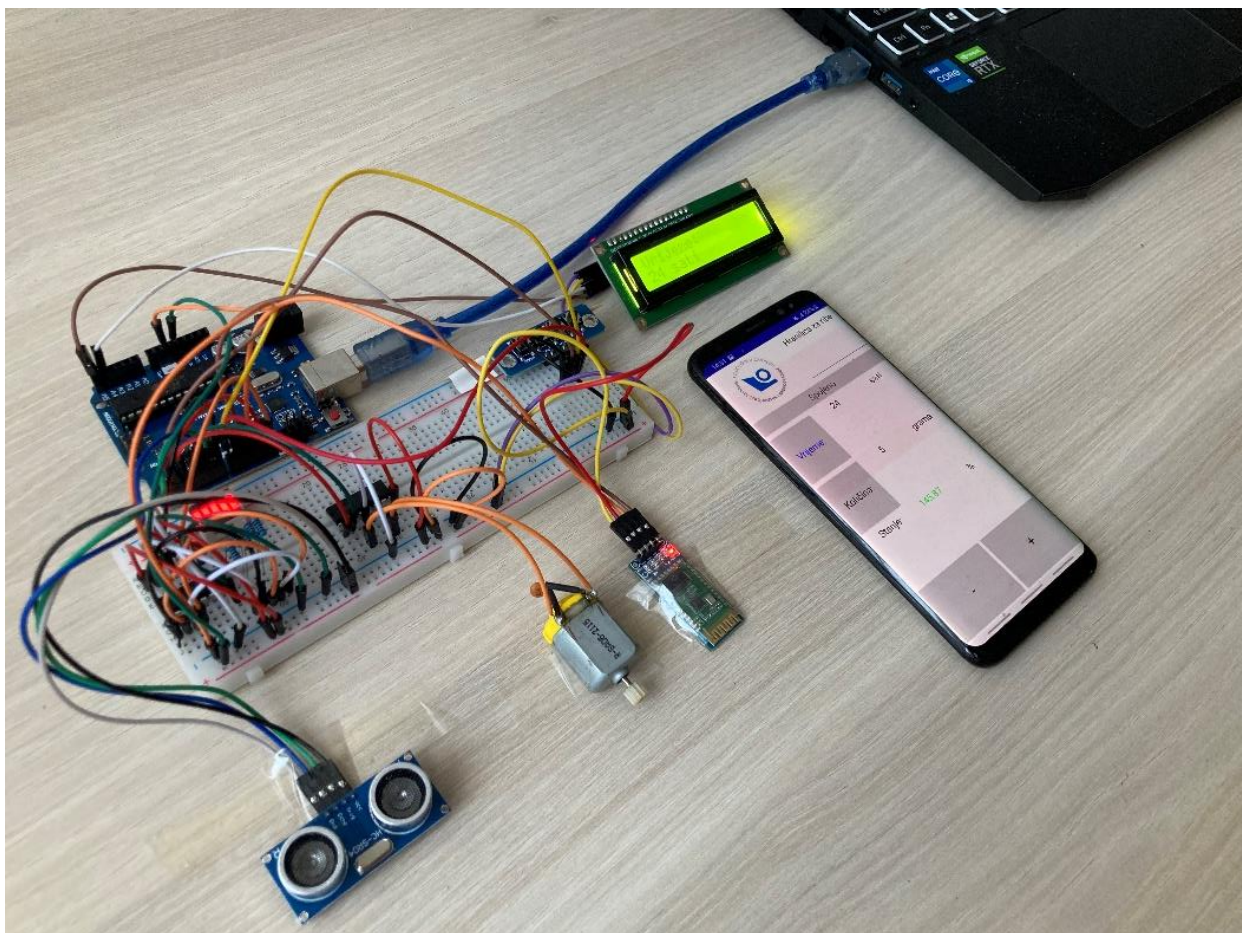
Na slici 5.1 dana je pojednostavljena električna shema koja na razumljiv način prikazuje raspored povezivanja električnih komponenti. Električne sheme su grafički prikazi koji koriste simbole za predstavljanje različitih elektroničkih komponenti, kao što su otpornici, kondenzatori, diode, tranzistori, senzori, mikroprocesori i drugi elementi. Cilj je jasno pokazati kako su komponente povezane u strujnom krugu i kako električna struja teče kroz njega. Ove sheme su ključne za inženjere i tehničare koji rade s elektronikom jer pružaju vizualni vodič koji pomaže u razumijevanju složenih sustava, olakšavaju otkrivanje problema te pružaju smjernice za montažu i testiranje. Električna shema izrađena je u programu kojemu se pristupa putem internetske stranice pod nazivom „Circuit Diagram Web Editor“.



Slika 5.1. Električna shema

5.2. Spajanje i testiranje (simulacija)

Da bi se sa sigurnošću znalo da konfiguracija komponenata međusobno odgovara tj. da su sve komponente međusobno kompatibilne te da teorijski ispravna el. shema radi u praksi, potrebno je provesti simulaciju rada sustava na stvarnom modelu (eksperiment). Budući da se sama hranilica u sklopu ovog rada nije izrađivala, simulacija je provedena s elektroničkim komponentama spojenim na eksperimentalnog pločici te s aplikacijom na mobilnom telefonu. Ovom simulacijom potvrđene su sve prethodne pretpostavke i utvrđen je ispravan rad sustava.



Slika 5.2. Eksperimentalni spoj



Slika 5.3. Primjer promjene boje prikaza promjenom stanja u spremniku hranilice

5.3. Diskusija

Volumenom spremnika hranu od 105 cm³ (ili 105 ml) omogućeno je skladištenje 26 grama hrane za ribice. Ako se uzme u obzir da se u akvariju nalazi broj ribica naveden u poglavljima 2.1.1. i 2.1.2., dolazi se do izračuna da hranilica može raditi potpuno autonomno oko tjedan dana. Taj broj naravno varira tako da bi u slučaju 2 ribice on iznosio čak 33 dana. Eksperimentima je utvrđeno da 9 V baterija koja bi napajala neprestano uključeni Arduino sustav srednje veličine trajala oko 8 dana, tako da je, uzevši u obzir faktor sigurnosti, zamjena baterije potrebna svakih 7 dana.

Kada se zbroji cijena svih elektroničkih komponenata dolazi se do cijene elektroničkog sklopa od oko 22 dolara, koja bi se povećala na oko 32 dolara kada bi se uračunala cijena izrade tiskane pločice za shift registar i motor kontroler. Budući da je predviđena tehnologija izrade 3D printanje, cijena filameta (materijal za printanje) od ABS plastike konstrukcije iznosila bi oko

7,5 dolara (masa konstrukcije iznosi oko 320 grama). Time se dolazi do ukupne cijene od oko 40 dolara koja ne uključuje radnu snagu potrebnu za izradu sklopa. Ako bi se odlučilo pokrenuti serijsku proizvodnju, cijena pojedinačnih komponenata i dijelova bi naravno bila nešto niža.

Na konstruiranje konstrukcije utrošeno je oko 30 radnih sati, na dizajn aplikacije oko 25 radnih sati dok je na programiranje utrošeno oko 40 radnih sati, što dovodi do zbroja od 95 radnih sati. Ako bi se realizirale izrada konstrukcije i tiskane pločice, bilo bi potrebno još oko 50 sati za prilagodbu komponenata (zračnosti, dosjedi itd.) i dizajn tiskane pločice.

Izradom ovog rada primijenjena su znanja stečena tokom prethodnog razdoblja studija mehatronike, poput programiranja, strojarstva i elektronike. Posjedovanje tih znanja ključno je za inženjera mehatronike budući da omogućavaju široki spektar vještina koje su potrebne inženjeru kako bi se mogao snaći u svim budućim izazovima.

6. ZAKLJUČAK

Automatska hranilica za akvarijske ribe omogućava lakšu brigu o ribama u akvariju te povećava njihovu sigurnost jer se eliminira faktor izostanka obroka. Svojim velikim kapacitetom posude za hranu te malom energetsom potrošnjom sustava omogućava se njen dugi autonomni rad. Upotrebom aplikacije na mobilnom uređaju (koja je poveziva s aplikacijom preko Bluetooth modula) dodatno se olakšava interakcija s hranilicom tj, podešavanje parametara na istoj.

Nedostatak sustava je mali domet Bluetooth modula čime se mogući prostor za upotrebu aplikacije smanjuje na 10-tak metara od hranilice te nedovoljan sustav upozorenja za signalizaciju korisniku o kritično niskom stupnju hrane u spremniku te istrošenosti baterije. To bi se moglo riješiti dodavanjem zvučnih signala za uzbunu te slanjem obavijesti putem mobilnog uređaja gdje god se korisnik nalazio. Također, moglo bi se uštediti puno prostora i dobiti više mogućnost kada bi se izradila prilagođena tiskana pločica na koju bi bio ugrađen veliki dio elektroničkih komponenata hranilice (Arduino, registar, motor kontroler...).

LITERATURA

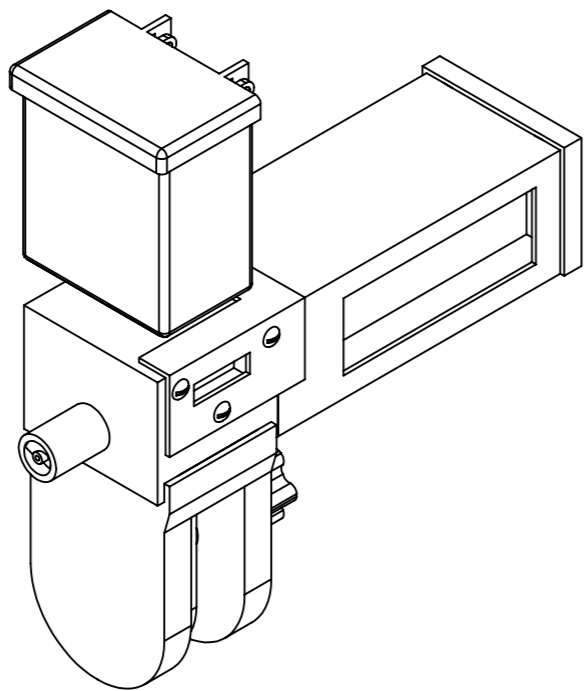
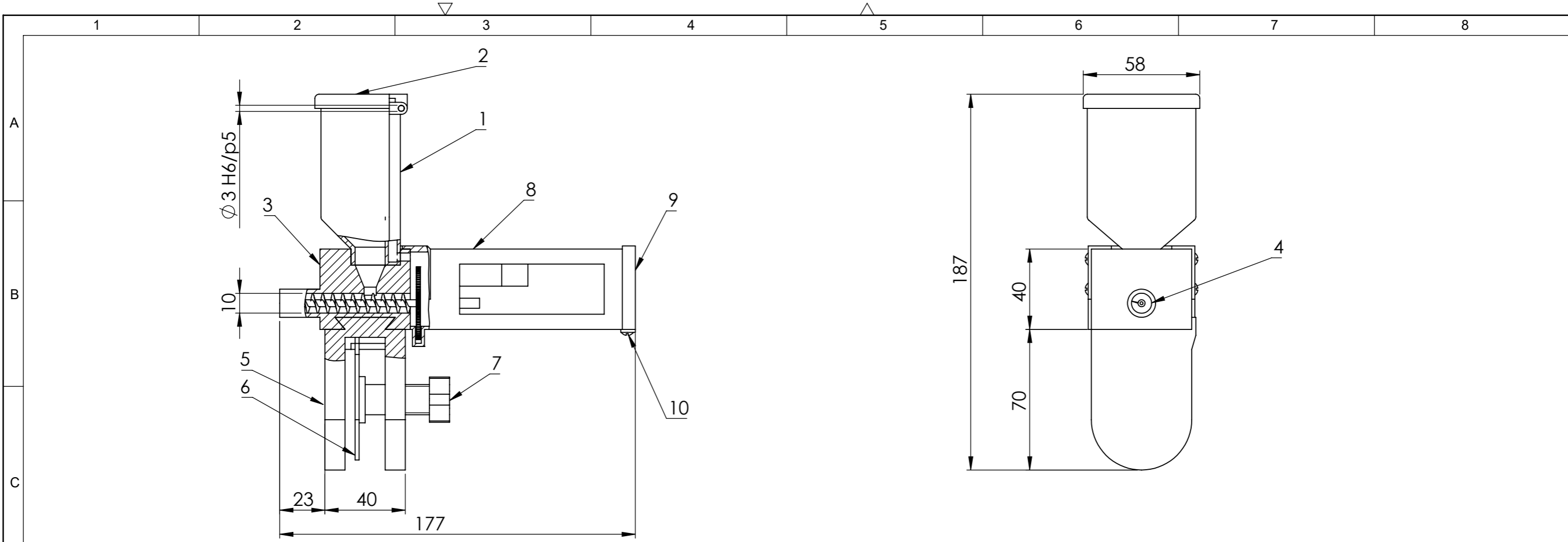
- [1] <https://www.quora.com/How-long-have-people-been-keeping-fish-in-aquariums-and-what-country-started-it-What-fish-were-used> (pristupljeno 1.8.2024.)
- [2] <https://sretniljubimci.com/najbolje-ribice-za-slatkovodni-akvarij/> (pristupljeno 6.8.2024.)
- [3] <https://www.pet-centar.hr/blogs/pitajte-veterinara/zlatne-ribice-za-pocetnike> (pristupljeno 6.8.2024.)
- [4] <https://goldfishkeepers.com/GoldfishFeedingGuide.php> (pristupljeno 6.8.2024.)
- [5] <https://grama.com.hr/category/domace-zivotinje-i-kucni-ljubimci/page/2/> (pristupljeno 8.9.2024.)
- [6] <https://www.pet-centar.hr/blogs/pitajte-veterinara/skalari> (pristupljeno 8.8.2024.)
- [7] <https://www.scielo.br/j/rbz/a/Gr3LBJ8GnmXGk78jhkbfzYB/?format=pdf&lang=en> (pristupljeno 9.8.2024.)
- [8] https://upload.wikimedia.org/wikipedia/commons/6/6e/Freshwater_angelfish_biodome.jpg (pristupljeno 8.9.2024.)
- [9] <https://www.pet-centar.hr/products/prodac-tropical-fish-flakes-hrana-u-listicima-za-sve-tropske-ribice?variant=47786041409812> (pristupljeno 11.8.2024.)
- [10] <https://www.njuskalo.hr/oprema-akvaristika/prodajem-akvarij-samo-staklo-80x35x40-oglas-14084085> (pristupljeno 8.9.2024.)
- [11] https://store.arduino.cc/en-hr/products/arduino-nano?srsId=AfmBOoqalGnogCFuc4iLzy9p3W31wz5N4jcLwu1IpXaxpW_p-5z2V1ID (pristupljeno 11.8.2024.)
- [12] https://www.welectronic.com/Arduino-Nano_1 (pristupljeno 8.9.2024.)
- [13] <https://soldered.com/hr/proizvod/ultrazvucni-modul-hc-sr04/> (pristupljeno 11.8.2024.)
- [14] https://www.waveshare.com/wiki/LCD1602_I2C_Module (pristupljeno 12.8.2024.)
- [15] <https://biopton.com.hr/branch-185045-original/> (pristupljeno 8.9.2024.)

-
- [16] <https://www.robotistan.com/hc06-serial-port-bluetooth-module> (pristupljeno 12.8.2024.)
- [17] <https://www.electronic.ba/bs/arduino-moduli/bluetooth-modul-arduino-hc-06rf> (pristupljeno 8.9.2024.)
- [18] https://www.aliexpress.com/item/33021641990.html?spm=a2g0o.productlist.main.13.53cd2279ZocjgX&algo_pvid=3e381533-2536-44ab-8082-de7f1999a910&algo_exp_id=3e381533-2536-44ab-8082-de7f1999a910-6&pdp_npi=4%40dis%21EUR%210.37%210.35%21%21%210.40%210.38%21%40211b653717240817958344406ef65d%2167306368638%21sea%21HR%212909579171%21X&curPageLogUid=4IGQeP8zSMBv&utparam-url=scene%3Asearch%7Cquery_from%3A#nav-specification (pristupljeno 14.8.2024.)
- [19] https://www.aliexpress.com/item/1005006327268446.html?spm=a2g0o.productlist.main.1.69d41411KJOx9H&algo_pvid=05f07b1a-9534-4114-973c-bf6a7ec41f78&algo_exp_id=05f07b1a-9534-4114-973c-bf6a7ec41f78-0&pdp_npi=4%40dis%21EUR%212.86%211.32%21%21%2121.99%2110.12%21%40211b617a17240874422758612e859c%2112000036774428452%21sea%21HR%212909579171%21X&curPageLogUid=pCh6j3mQYFSK&utparam-url=scene%3Asearch%7Cquery_from%3A#nav-specification (pristupljeno 16.8.2024.)
- [20] https://www.aliexpress.com/item/1005004488935863.html?spm=a2g0o.productlist.main.25.6cda239dENCOPy&algo_pvid=8e66a4b7-7960-44fb-ac56-d122bfdd2891&algo_exp_id=8e66a4b7-7960-44fb-ac56-d122bfdd2891-12&pdp_npi=4%40dis%21EUR%211.87%211.76%21%21%2121.01%211.89%21%40211b80c217240881779496621e9cd4%2112000029334924568%21sea%21HR%212909579171%21X&curPageLogUid=c9Pn1h8LPA4E&utparam-url=scene%3Asearch%7Cquery_from%3A#nav-specification (pristupljeno 16.8.2024.)
- [21] <https://www.duracell.com/wp-content/uploads/2020/02/9V-Duracell-Plus.pdf> (pristupljeno 19.8.2024.)
- [22] <https://dmm-unik.hr/proizvod/baterija-duracell?var=i100653> (pristupljeno 8.9.2024.)
- [23] <https://olx.ba/artikal/59042456> (pristupljeno 8.9.2024.)
- [24] <https://www.ti.com/product/L293D#features> (pristupljeno 20.8.2024.)
-

[25] https://www.researchgate.net/figure/L293D-Motor-Driver-IC-Actuator-Force-Calculation-The-following-arrangement-was-made-to_fig1_313625689 (pristupljeno 8.9.2024.)

[26] <https://hr.mouser.com/ProductDetail/Texas-Instruments/SN74HC595N?qs=IEI3ej0IqwBTHkYa8XPoMQ%3D%3D> (pristupljeno 20.8.2024.)

[27] <https://www.instructables.com/3-Wire-HD47780-LCD-for-less-than-1-dollar/> (pristupljeno 8.9.2024.)



Izometrijski prikaz
M 1:2

1	Spremnik za hranu	1			50 x 43 x 83
2	Poklopac spremnika za hranu	1			58 x 46 x 10.5
3	Glavna nosiva konstrukcija	1			65 x 50 x 40
4	Vijak za dobavu hrane	1			Ø36.2 x 70
5	Tijelo škripca	1			40 x 52 x 76
6	Pritisna ploha škripca	1			5 x 40 x 61
7	Vijak za zatezanje	1			Ø25 x 45
8	Kućište	1			152.5 x 50 x 41.5
9	Poklopac kućišta	1			7.25 x 53 x 43
10	Vijak	7		DIN 13 T1	M2 x 0.4

Poz.	Naziv dijela	Kom.	Crtež broj Norma	Materijal	Sirove dimenzije Proizvođač	Masa
Broj naziva - code		Datum	Ime i prezime	Potpis		
Projektirao		1.9.2024.	Karlo Martić			
Razradio		1.9.2024.	Karlo Martić			
Crtao		1.9.2024.	Karlo Martić			
Pregledao			Mihael Cipek			
Mentor						
ISO - tolerancije		Objekt:		Objekt broj:		
Ø3 H6/p5	0,017 0			R. N. broj:		
		Napomena:		Smjer: MiR		Kopija
		Materijal:	Masa:	ZAVRŠNI RAD		
		Naziv: Automatizirana akvarijska hranilica		Pozicija:		
		Mjerilo originala M 1:2		Format: A3		Listova: 1
		Crtež broj: 01_010924_KM		List: 1		

